jc966 U.S. PTO
11/20/00

11-21-00

Please type a plus sign (+) inside this box → [+]

PTO/SB/05 (12/97)
Approved for use through 09/30/00. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	*GEMS:0131/yod 15-IS-5887*	*Total Pages*	*84*
First Named Inventor or Application Identifier	*Robert D. Barnes*		
Express Mail Label No.	***EL 652 334 409 US***		

jc930 U.S. PTO
09/716603
11/20/00

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: **Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231**

1. ☒ Fee Transmittal Form *(Submit an original, and a duplicate for fee processing)*
2. ☒ Specification *Total Pages* ***39***
   (preferred arrangement set forth below)
   -Descriptive
   -Cross References to Related Application
   -Statement Regarding Fed sponsored R & D
   -Reference to Microfiche Appendix
   -Background of the Invention
   -Brief Summary of the Invention
   -Brief Description of the Drawings *(if filed)*
   -Detailed Description
   -Claim(s)
   -Abstract of the Disclosure
3. ☒ Drawing(s) *(35 USC 113)* *Total Sheets* ***12*** *Total Pages* ***36***
4. Oath or Declaration
   a. ☒ Newly executed (original or copy)
   b. ☐ Copy from a prior application (37CFR 1.63(d))
   (for continuation/divisional with Box 17 completed)
   [Note Box 5 below]
   i. ☐ DELETION OF INVENTOR(S)
   Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
5. ☐ Incorporation By Reference *(useable if Box 4b is checked)*
   The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.
6. ☐ Microfiche Computer Program *(Appendix)*
7. Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
   a. ☐ Computer Readable Copy
   b. ☐ Paper Copy (identical to computer copy)
   c. ☐ Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. ☒ Assignment Papers (cover sheet & document(s))
9. ☐ 37 CFR 3.73(b) Statement ☒ Power of Attorney *(where there is an assignee)*
10. ☐ English Translation Document *(if applicable)*
11. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations
12. ☐ Preliminary Amendment
13. ☒ Return Receipt Postcard (MPEP 503)
14. ☐ Small Entity Statement(s) ☐ Statement filed in prior application Status still proper and desired
15. ☐ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*
16. ☐ Other

17. ☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No: ____/____

18. CORRESPONDENCE ADDRESS

☐ *Customer Number or Bar Code Label*

(Insert Customer No. or Attach bar code label here)

☒ *Correspondence address below*

NAME	Patrick S. Yoder				
	Fletcher, Yoder & Van Someren				
ADDRESS	P.O. Box 692289				
CITY	Houston	*STATE*	Texas	*ZIP CODE*	77269-2289
COUNTRY	USA	*TELEPHONE*	(281) 970-4545	*Fax*	(281) 970-4503

Burden Hour Statement. This form is estimated to take 0.2 hours to complete Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231 DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

PTO/SB/17 (10/96)
Approved for use through 09/30/98. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

11/20/00 JC788 U.S. PTO

FEE TRANSMITTAL

Complete if Known	
Application Number	unassigned
Filing Date	herewith
First Named Inventor	Robert D. Barnes
Group Art Unit	unknown
Examiner Name	unknown
Attorney Docket Number	GEMS:0131/YOD (15-IS-5887)

TOTAL AMOUNT OF PAYMENT | ($) 750.00

METHOD OF PAYMENT (check one)

1. ☒ The Commissioner is hereby authorized to charge indicated fees and credit any over payments to:

Deposit Account Number: 07-0845/GEMS:0131/YOD (15-IS-5887)

Deposit Account Name: GE Medical Systems

☒ Charge Any Additional Fee Required Under 37 CFR 1 16 and 1.17

☐ Charge the Issue Fee Set in 37 CFR 1.18 at the Mailing of the Notice of Allowance, 37 CFR 1.31(b)

2. ☐ Payment Enclosed:

☐ Check ☐ Money Order ☐ Other

FEE CALCULATION (fees effective 10/01/96)

1. FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
101	690	201	395	Utility filing fee	710 00
106	330	206	165	Design filing fee	___
107	540	207	270	Plant filing fee	___
108	790	208	395	Reissue filing fee	___
114	150	214	75	Provisional filing fee	___
				SUBTOTAL (1)	($) 710.00

2. CLAIMS

		Extra		Fee from below		Fee Paid
Total Claims	1 -20 =	0	X	18	=	0.00
Independent Claims	1 - 3 =	0	X	80	=	0.00
Multiple Dependent Claims		___	X	___	=	___

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
103	18	203	11	Claims in excess of 20
102	78	202	41	Independent claims in excess of 3
104	270	204	135	Multiple dependent claim
109	82	209	41	Reissue independent claims over original patent
110	22	210	11	Reissue claims in excess of of 20 and over original patent
				SUBTOTAL (2) ($) 0.00

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Fee Code	Large Entity Fee ($)	Small Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	___
127	50	227	25	Surcharge - late provisional filing or cover sheet.	___
139	130	139	130	Non-English specification	___
147	2,520	147	2,520	For filing a request for reexamination	___
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	___
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	___
115	110	215	55	Extension for response within first month	___
116	400	216	200	Extension for response within second month	___
117	950	217	475	Extension for response within third month	___
118	1,570	218	755	Extension for response within fourth month	___
119	310	219	155	Notice of Appeal	___
120	310	220	155	Filing a brief in support of an appeal	___
121	270	221	135	Request for oral hearing	___
138	1,510	138	1,510	Petition to institute a public use proceeding	___
140	110	240	55	Petition to revive unavoidably abandoned application	___
141	1,320	241	660	Petition to revive unintentionally abandoned application	___
142	1,320	242	660	Utility issue fee (or reissue)	___
143	450	243	225	Design issue fee	___
144	670	244	335	Plant issue fee	___
122	130	122	130	Petitions to the Commissioner	___
123	50	123	50	Petitions related to provisional applications	___
126	240	126	240	Submission of Information Disclosure Stmt	___
581	40	581	40	Recording each patent assignment per property (times number of properties)	40 00
146	790	246	395	Filing a submission after final rejection (37 CFR 1 129(a))	___
149	790	249	395	For each additional invention to be examined (37 CFR 1.129(b))	___
Other fee (specify)					___
Other fee (specify)					___
				SUBTOTAL (3)	($) 40.00

* Reduced by Basic Filing Fee Paid

SUBMITTED BY					Complete (if applicable)
Typed or Printed Name	Patrick S Yoder	Reg. Number	37,479		
Signature		Date	November 20, 2000	Deposit Acct User ID	07-0845/GEMS 0131 15-IS-5887

11/27/2000 VTANG1 00000004 070845 09716603
Sale Ref: 00000004 DAN: 070845 09716603
01 FC:101 710.00 CH

15-IS-5887

IMAGE DATA COMPRESSION METHOD AND APPARATUS

by

Robert D. Barnes
Saad A. Sirohey
Robert C. Gemperline

EXPRESS MAIL MAILING LABEL NUMBER: EL 652 334 093 US
DATE OF DEPOSIT: November 20, 2000

Pursuant to 37 C.F.R. § 1.10, I hereby certify that I am personally depositing this paper or fee with the U.S. Postal Service, "Express Mail Post Office to Addressee" service on the date indicated above in a sealed envelope (a) having the above-numbered Express Mail label and sufficient postage affixed, and (b) addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

November 20, 2000 — Date
Lynda Howell — Lynda Howell

IMAGE DATA COMPRESSION METHOD AND APPARATUS

CROSS REFERENCE TO RELATED APPLICATIONS

The present is a Continuation-In-Part of Patent Application Serial No. 09/448,950, filed on November 24, 1999.

FIELD OF THE INVENTION

The present invention relates generally to the field of image data compression. More particularly, the invention relates to a technique for compressing image data for rapid transmission and decompression.

BACKGROUND OF THE INVENTION

A wide range of applications exist for image data compression. Digitized images may be created in a variety of manners, such as via relatively simple digitizing equipment and digital camera, as well as by complex imaging systems, such as those used in medical diagnostic applications. Regardless of the environment in which the image data originates, the digital data descriptive of the images is stored for later reconstruction and display, and may be transmitted to various locations by networks, such as the Internet. Goals in digital image management include the efficient use of memory allocated for storage of the image data, as well as the efficient and rapid transmission of the image data for reconstruction. The latter goal is particularly important where large or complex images are to be handled over comparatively limited bandwidth networks. In the medical diagnostic imaging field, for example, very large image data sets may be available for transmission and viewing by a range of users, including those having limited access to very high bandwidths needed for rapid transmission of full detail images.

Picture archiving and communication systems, or PACS, have become an extremely important component in the management of digitized image data,

particularly in the field of medical imaging. Such systems often function as central repositories of image data, receiving the data from various sources, such as medical imaging systems. The image data is stored and made available to radiologists, diagnosing and referring physicians, and other specialists via network links. Improvements in PACS have led to dramatic advances in the volumes of image data available, and have facilitated loading and transferring of voluminous data files both within institutions and between the central storage location or locations and remote clients.

A major challenge to further improvements in all image handling systems, from simple Internet browsers to PACS in medical diagnostic applications, is the handling of the large data files defining images. In the medical diagnostics field, depending upon the imaging modality, digitized data may be acquired and processed for a substantial number of images in a single examination, each image representing a large data set defining discrete picture elements or pixels of a reconstructed image. Computed Tomography (CT) imaging systems, for example, can produce numerous separate images along an anatomy of interest in a very short examination timeframe. Ideally, all such images are stored centrally on the PACS, and made available to the radiologist for review and diagnosis.

Various techniques have been proposed and are currently in use for analyzing and compressing large data files, such as medical image data files. Image data files typically include streams of data descriptive of image characteristics, typically of intensities or other characteristics of individual pixels in the reconstructed image. In the medical diagnostic field, these image files are typically created during an image acquisition or encoding sequence, such as in an x-ray system, a magnetic resonance imaging system, a computed tomography imaging system, and so forth. The image data is then processed, such as to adjust dynamic ranges, or to enhance certain features shown in the image, for storage, transmittal and display.

While image files may be stored in raw and processed formats, many image files are quite large, and would occupy considerable disc or storage space. The increasing complexity of imaging systems also has led to the creation of very large image files, typically including more data as a result of the useful dynamic range of the imaging system, the size of the matrix of image pixels, and the number of images acquired per examination.

In addition to occupying large segments of available memory, large image files can be difficult or time consuming to transmit from one location to another. In a typical medical imaging application, for example, a scanner or other imaging device will typically create raw data which may be at least partially processed at the scanner. The data is then transmitted to other image processing circuitry, typically including a programmed computer, where the image data is further processed and enhanced. Ultimately, the image data is stored either locally at the system, or in the PACS for later retrieval and analysis. In all of these data transmission steps, the large image data file must be accessed and transmitted from one device to another.

Current image handling techniques include compression of image data within the PACS environment to reduce the storage requirements and transmission times. Such compression techniques may, however, compress entire files, including descriptive header information which could be useful in accessing or correlating images for review. Moreover, current techniques do not offer sufficiently rapid compression and decompression of image files to satisfy increasing demands on system throughput rates and access times. Finally, alternative compression and decompression techniques do not offer the desired compression ratios, in combination with rapid compression and decompression in a client-server environment.

Another drawback of existing compression techniques is the storage, access and transmission of large data files even when a user cannot or does not desire to view the reconstructed image in all available detail. For example, in medical

imaging, extremely detailed images may be acquired and stored, while a radiologist or physician who desires to view the images may not have a view port capable of displaying the image in the resolution in which they are stored. Thus, transmission of the entire images to a remote viewing station, in relatively time consuming operations, may not provide any real benefit and may slow reading or other use of the images.

There is a need, therefore, for an improved image data compression and decompression technique which provides rapid compression and decompression of image files, and which obtains improved compression ratios and transmission times. There is a particular need for a technique which permits compressed image data files to be created and transmitted in various resolutions or sizes, depending upon the bandwidth and desired or available resolution on a client side.

SUMMARY OF THE INVENTION

The invention provides an improved image data compression technique designed to respond to these needs. The system makes use of extremely fast compression and decompression techniques to create compressed image data files. The compression proceeds based upon lossless integer wavelet decomposition of the image data, followed by adaptive Huffman code compression of the resulting data sets. The decomposition may continue through a series of successive steps, each providing progressively smaller images without significant loss of image detail. A user may then selectively access an appropriate data set for image reconstruction in accordance with available bandwidth, viewing resolution and time constraints. Increasingly detailed images can then be accessed by accessing a reduced number of additional data sets for detail in a larger or higher resolution image. The technique thus provides a very rapid and efficient lossless compression and decompression approach well suited to a range of applications, including medical diagnostic imaging.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagrammatical representation of an exemplary image management system, in the illustrated example a picture archiving and communication system or PACS, for receiving and storing image data in accordance with certain aspects of the present technique;

Fig. 2 is a diagrammatical representation of contents of a database for referencing stored image data in files containing multiple image data sets, compressed data, and descriptive information;

Fig. 3 is a representation of a typical image of the type received, compressed, and stored on the system of Fig. 1;

Fig. 4 is a graphical representation of intensities of pixelated data across an image, subdivided into subregions for compression of the subregions optimally based upon characteristics of the subregions;

Fig. 5 is a diagrammatical representation of a pixel neighborhood used in the analysis of the image data for compression purposes;

Fig. 6 is a flow chart illustrating exemplary control logic for compressing and decompressing image data in accordance with aspects of the present technique;

Figs. 7, 8, 9, 10, 11 and 12 are exemplary lookup tables in the form of compression code tables used to optimally compress subregions of image data in accordance with the present technique during the process illustrated in Fig. 6;

Fig. 13 is a diagrammatical representation of an exemplary image data set, including a descriptive header, a compression header, and blocks of compressed data by subregion;

Fig. 14 is a process map illustrating progressive wavelet decomposition of image data in accordance with a further aspect of the present technique, permitting generation of a multiresolution compressed image data set;

Fig. 15 is a process map illustrating the progressive wavelet decomposition of Fig. 14 in conjunction with compression of the resulting data for compression of generation of the multiresolution compressed image data set;

Fig. 16 is a flow chart illustrating exemplary logic in performing forward wavelet transformation in accordance with a present embodiment;

Fig. 17 is a flow chart illustrating exemplary logic in performing forward wavelet decomposition for multiresolution image data compression;

Fig. 18 is a diagrammatical representation of an exemplary image data set including a header and blocks of compressed data in multiresolution compression;

Fig. 19 is a process map illustrating image data decompression and inverse wavelet transformation to obtain images of desired resolution; and

Fig. 20 is a table indicating exemplary compression levels obtained via the present technique.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 illustrates an exemplary image data management system in the form of a picture archive and communication system or PACS 10 for receiving, compressing and decompressing image data. In the illustrated embodiment, PACS 10 receives image data from several separate imaging systems designated by reference numerals 12, 14 and 16. As will be appreciated by those skilled in the art, the imaging systems may be of various type and modality, such as magnetic resonance imaging (MRI) systems, computed tomography (CT) systems, positron emission tomography (PET) systems, radio fluoroscopy (RF), computed radiography (CR), ultrasound systems, and so forth. Moreover, the systems may include processing stations or digitizing stations, such as equipment designed to provide digitized image data based upon existing film or hard copy images. It should also be noted that the systems supplying the image data to the PACS may be located locally with respect to the PACS, such as in the same institution or facility, or may be entirely remote from the PACS, such as in an outlying clinic or affiliated institution. In the latter case, the image data may be transmitted via any suitable network link, including open networks, proprietary networks, virtual private networks, and so forth.

PACS 10 includes one or more file servers 18 designed to receive and process image data, and to make the image data available for decompression and

review. Server 18 receives the image data through an input/output interface 19. Image data may be compressed in routines accessed through a compression/decompression interface 20. As described more fully below, interface 20 serves to compress the incoming image data rapidly and optimally, while maintaining descriptive image data available for reference by server 18 and other components of the PACS. Where desired, interface 20 may also serve to decompress image data accessed through the server. The server is also coupled to internal clients, as indicated at reference numeral 22, each client typically including a work station at which a radiologist, physician, or clinician may access image data from the server, decompress the image data, and view or output the image data as desired. Clients 22 may also input information, such as dictation of a radiologist following review of examination sequences. Similarly, server 18 may be coupled to one or more interfaces, such as a printer interface 24 designed to access and decompress image data, and to output hard copy images via a printer 26 or other peripheral.

Server 28 may associate image data, and other work flow information within the PACS by reference to one or more file servers 18. In the presently contemplated embodiment, database server 28 may include cross-referenced information regarding specific image sequences, referring or diagnosing physician information, patient information, background information, work list cross-references, and so forth. The information within database server 28 serves to facilitate storage and association of the image data files with one another, and to allow requesting clients to rapidly and accurately access image data files stored within the system. Similarly, server 18 is coupled to one or more archives 30, such as an optical storage system, which serve as repositories of large volumes of image data for backup and archiving purposes. Techniques for transferring image data between server 18, and any memory associated with server 18 forming a short term storage system, and archive 30, may follow any suitable data management scheme, such as to archive image data following review and dictation by a radiologist, or after a sufficient time has lapsed since the receipt or review of the image files.

In the illustrated embodiment, other components of the PACS system or institution may be integrated with the foregoing components to further enhance the system functionality. For example, as illustrated in Fig. 1, a compression/decompression library 32 is coupled to interface 20 and serves to store compression routines, algorithms, look up tables, and so forth, for access by interface 20 (or other system components) upon execution of compression and decompression routines (i.e. to store various routines, software versions, code tables, and so forth). In practice, interface 20 may be part of library 32. Library 32 may also be coupled to other components of the system, such as client stations 22 or printer interface 24, serving similarly as a library or store for the compression and decompression routines and algorithms. Although illustrated as a separate component in Fig. 1, it should be understood that library 32 may be included in any suitable server or memory device, including within server 18. Moreover, code defining the compression and decompression processes described below may be loaded directly into interface 20 and/or library 32, or may be loaded or updated via network links, including wide area networks, open networks, and so forth.

Additional systems may be linked to the PACS, such as directly to server 28, or through interfaces such as interface 19. In the embodiment illustrated in Fig. 1, a radiology department information system or RIS 34 is linked to server 18 to facilitate exchanges of data, typically cross-referencing data within database server 28, and a central or departmental information system or database. Similarly, a hospital information system or HIS 36 may be coupled to server 28 to similarly exchange database information, workflow information, and so forth. Where desired, such systems may be interfaced through data exchange software, or may be partially or fully integrated with the PACS system to provide access to data between the PACS database and radiology department or hospital databases, or to provide a single cross-referencing database. Similarly, external clients, as designated at reference numeral 38, may be interfaced with the PACS to enable images to be viewed at remote locations. Such external clients may employ decompression software, or may receive image files already decompressed by interface 20. Again,

links to such external clients may be made through any suitable connection, such as wide area networks, virtual private networks, and so forth.

Fig. 2 illustrates in somewhat greater detail the type of cross-referencing data made available to clients 20, 22, 24, 30 through database server 28. The database entries, designated generally by reference numeral 40 in Fig. 2, will include cross-referenced information, including patient data 42, references to specific studies or examinations 43, references to specific procedures performed 44, references to anatomy imaged 45, and further references to specific image series 46 within the study or examination. As will be appreciated by those skilled in the art, such cross-referenced information may include further information regarding the time and date of the examination and series, the name of diagnosing, referring, and other physicians, the hospital or department where the images are created, and so forth. The database will also include address information identifying specific images, file names, and locations of the images as indicated at reference numeral 48. Where the PACS includes various associated memory devices or short term storage systems, these locations may be cross-referenced within the database and may be essentially hidden from the end user, the image files simply being accessed by the system for viewing from the specific storage location based upon cross-referenced information in the database.

As described more fully below, in accordance with certain aspects of the present technique, descriptive information is used to identify preferred or optimal compression routines used to compress image data. Such descriptive information is typically available from header sections of an image data string, also as described in detail below. However, information available from database server 28 may also serve as the basis for certain of the selections of the algorithms employed in the compression technique. Specifically database references may be relied upon for identifying such descriptive information as the procedures performed in an imaging sequence, specific anatomies or other features viewable in reconstructed images

based upon the data, and so forth. Such information may also be available from the RIS 34 and from the HIS 36.

Fig. 2 also illustrates an exemplary image file cross-referenced by the database entries. As shown in Fig. 2, image file 50 includes a plurality of image data sets 52, 54 and 56. In a typical image file, a large number of such image sets may be defined by a continuous data stream. Each data set may be compressed in accordance with specific compression algorithms, including the compression algorithms as described below.

Within each image data set, a descriptive header 58 is provided, along with a compression header 60. The headers 58 and 60 are followed by compressed image data 62. The descriptive header 58 of each data set preferably includes industry-standard or recognizable descriptive information, such as DICOM compliant descriptive data. As will be appreciated by those skilled in the art, such descriptive information will typically include an identification of the patient, image, date of the study or series, modality of the system creating the image data, as well as additional information regarding specific anatomies or features visible in the reconstructed images. As described more fully below, such descriptive header data is preferably employed in the present technique for identification of optimal compression algorithms or routines used to compress the data within the compressed image data section 62. Data referring to the specific algorithm or routine used to compress the image data is then stored within compression header 60 for later reference in decompressing the image data. As described below, additional data is stored within the compressed image data, cross-referencing the algorithms identified in compression header 60 for use in decompressing the image data. Specifically, in a presently preferred embodiment, the compression header 60 includes identification of the length of subregions of the compressed image data, as well as references to specific optimal algorithms, in the form of compression code tables used to compress the subregions optimally.

Fig. 3 illustrates a typical image which is encoded by packets of digitized data assembled in a continuous data stream which may be compressed and decompressed in the present techniques. The image, designated generally by the reference numeral 100, will typically include features of interest 102, such as specific anatomical features. In medical diagnostic applications, such features may include specific anatomies or regions of a patient viewable by virtue of the physics of the image acquisition modality, such as soft tissue in MRI system images, bone in x-ray images, and so forth. Each image is comprised of a matrix having a width 104 and a height 106 defined by the number and distribution of individual pixels 108. The pixels of the image matrix are arranged in rows 110 and columns 112, and will have varying characteristics which, when viewed in the reconstructed image, define the features of interest. In a typical medical diagnostic application, these characteristics will include gray level intensity or color. In the digitized data stream, each pixel is represented by binary code, with the binary code being appended to the descriptive header to aid in identification of the image and in its association with other images of a study. As noted above, such descriptive information may include industry standard information, such as DICOM compliant data.

Fig. 4 graphically represents intensities of pixel data defining an image across a pair of rows of the image matrix. Each row of the image matrix will include a series of pixels, each pixel being encoded by binary data descriptive of the pixel characteristics, typically intensity. Thus, lighter regions of the reconstructed image will correspond to pixels having a higher intensity level, with darker regions having a lower intensity level. When illustrated graphically, the intensity levels of the pixels across the image may form a contour or trace as shown in Fig. 4. Specifically, Fig. 4 illustrates a first row 114 lying adjacent to a second row 116, each row including a series of pixels having various intensities as indicated by traces 118 and 120, respectively. As will be appreciated by those skilled in the art, in practice, the graph of the pixel intensities would form a step-wise function along the position axis 122 and having amplitudes varying along the intensity axis 124.

It may be noted from Fig. 4 that in an actual image, variations in intensity along rows, and along columns as represented by correspondingly located intensities moving downwardly or upwardly through adjacent rows, will vary in accordance with the features represented in the reconstructed image. As illustrated in Fig. 4, row 114 includes areas of rising and declining intensity, including areas of both low intensity and high intensity. Row 116 includes areas of similar intensity, but varying by virtue of the features represented in the image. In accordance with the present technique, the image data stream is reduced to subregions as represented generally by reference numeral 126 in Fig. 4. While the subregions may be of different lengths (i.e. numbers of pixels), in the presently preferred embodiment, each subregion includes data encoding an equal number of pixels. Those skilled in the art will readily recognize, however, that after compression the actual length of codes for the subregion will vary depending upon the intensity of the pixels within the subregion and the dynamic range of the digital data encoding the pixel intensities. It should also be noted that where the row length of the image matrix is an integer multiple of the subregion width, individual subregions will align with one another moving down the image matrix as represented in Fig. 4. In general, however, the present technique is not limited to such integer multiple row widths.

Each subregion of the image data stream may be analyzed to identify entropy levels of the image data for compression purposes. In general, the entropy of the image data refers to the relative variation in the pixel intensities within each subregion. Thus, although a specific subregion may include pixels of a high intensity or a low intensity, where the intensity level is relatively stable or constant in the subregion, entropy is considered to be low. Such regions are shown in Fig. 4, for example, in portions of traces 114 and 116 indicated by reference numerals 128 and 132, respectively. On the contrary, where substantial variations in pixel intensity are encoded by the subregion data, the entropy level is considered high. Such regions are shown in Fig. 4 at reference numerals 130 and 134. It has been found that by subdividing the image data stream into subregions, which may have various lengths depending upon the image characteristics, and by analyzing the

relative entropy level of each subregion, the subregions may be optimally compressed in accordance with one of several candidate compression algorithms. Specifically, the present technique employs a series of predefined compression code tables which serve to translate pixel prediction errors into compressed codes for each subregion. The specific compression code table selected for each subregion is a function of the entropy identified for the subregion in the compression routine.

As will be described more fully below, the present technique preferably employs pixel value predictors and identifies differences between predicted values of individual pixels (i.e. binary code for the pixel intensity or other characteristic) and the actual value for the respective pixels. In fact, several predictor algorithms may be employed, with the specific predictor being identified based upon image characteristics, such as characteristics encoded in a descriptive header for the image data stream. The predictors are based upon comparisons of target pixels, or pixels of interest, with neighboring pixels. Fig. 5 represents a pixel neighborhood which serves as the basis for references made in the predictor algorithms. The pixel neighborhood, identified by reference numeral 138 in Fig. 5, includes the pixel of interest 136, designated p(i, j). Neighboring pixels include a "northwest" pixel 140, designated p(i-1, j-1), a "north" pixel 142 designated p(i, j-1), a "northeast" pixel 144 designated p(i+1, j-1), a "west" pixel 146 designated p(i-1, j). Some or all of these pixels may be employed in the predictor algorithm, or other additional pixels may be provided in a larger neighborhood.

The preferred technique for compressing the image data stream in the system described above to create hybrid compressed image data files is summarized in Fig. 6. The control logic may be subdivided into a series of logical blocks or segments, including a configuration segment 250 in which optimal compression algorithms, tables, predictor preferences, block or subregion length preferences, and so forth, are established. In a data access and retrieval segment 252 the image data to be compressed is received, as is descriptive information from database server 28, if used. In an evaluation segment 254 the images are evaluated based upon the

algorithms and preferences established in configuration segment 250. In a compression segment 256 certain values are computed and selected for the compression routine, and the image data is compressed to create the resulting compressed file. File code is added to the compressed data in a final segment 258. The file is then completed at step 260 and stored at step 262. The compression and decompression logic is then completed by eventual access of the compressed image file at step 264, retrieval of a decompression algorithm or algorithms at step 266 and decompression at step 268 for viewing and output. The foregoing logical segments and processes will be described below in greater detail.

The configuration segment 250 of the control logic includes steps of configuring the specific compression, predictor and block or subregion algorithms employed in the routine. Thus, at step 270 a series of compression code tables are generated for optimally compressing subregions of image data based upon relative entropy levels as indicated by prediction errors. As will be appreciated by those skilled in the art, such compression code tables serve to cross-reference original values with compressed values, generally in accordance with anticipated distributions or frequencies of occurrence of the original values. In the presently preferred embodiment, a series of compression code tables are established based upon analysis of typical images to be compressed by the routine. While the specific coding and ranges implemented in the compression code tables vary, and will generally be determined in specific applications empirically, examples of several such compression code tables are illustrated in Figs. 7 – 12.

Referring to Figs. 7 – 12, each compression code table, such as table 170 in Fig. 7, comprises a series of compressed data values 172 cross-referenced to original image parameters 174. In the example illustrated in the Figures, analysis may include an indication of the resulting code links in bits, as indicated by reference numeral 176, and the span of code translated by each table entry, as indicated by reference numeral 178. In the illustrated embodiment, the compressed data code of column 172 translates difference values of column 174 as identified by application

of predictors in the compression routine, followed by determination of the differences between the predicted and actual values. Thus, by way of example, where a difference value of 1 is identified based upon a predictor algorithm as described below, table 170 provides a compressed data code of 100. Again, it should be noted that the difference value does not generally represent the value of the encoded pixel itself, but a difference between a predicted value and an actual value. Thus, in a code length of 3 bits, summarized in column 176 of table 170, the pixel of interest differing from the predicted value by 1 will obtain a compressed data value of 100, although the pixel value itself may be considerably longer. As also may be seen in Fig. 7, the compressed data codes provided in column 172 may be summarized as including a first portion 180 designating the level within the table, followed by a second portion 182 which indicates the position within the level or range. Thus, for example, in the case of the table of Fig. 7, a difference range of from –2 to –3 would be encoded as 1101 followed by an additional bit set to a value of 0 or 1 depending upon whether the difference is –2 or –3. At an upper limit of the range, the compressed data code is taken to be the actual value of the individual pixel as described below with reference to Fig. 13.

As can be seen from the Figures, the compression code tables for translating prediction errors or differences to compressed code are established to provide a range of coding appropriate to the levels or variations in the difference values for each subregion of the data stream. Specifically, table 170 of Fig. 7 is adapted to a lowest entropy level as indicated by the low difference variation (zero) accommodated by the shortest code in the table and the relatively fine steps between the difference levels. Fig. 8 represents a second compression code table 184 providing for relatively higher entropy as indicated by the relatively broader maximum levels accommodated by the table and the relatively higher difference ranges as compared to table 170 of Fig. 7. Figs. 9, 10, 11 and 12 provide additional examples of compression code tables 186, 188, 190 and 192, respectively, for encoding successively higher entropy levels as indicated by prediction errors or differences within the various subregions. In the present embodiment illustrated, the

code tables are constructed using a Huffman code-based prefix and a multibit extension.

It should be noted that the compression ratio provided by each of the compression code tables in the family of tables varies depending upon the relative entropy level to be accommodated. Thus, table 170 of Fig. 7 provides a relatively high maximum compression ratio of 16:1, while table 192 of Fig. 12 provides a lower maximum compression ratio of 2.29:1. However, it has been found that the use of a plurality of different compression code tables to accommodate the different entropies of the data stream subregions results in excellent compression due to the use of compressed data code lengths optimally selected to accommodate the variations in prediction differences for the subregions.

It has been found that for specific types of images or for images having specific typical characteristics, various entropy levels may be anticipated. For example, in medical diagnostic imaging, relatively high entropy levels may be expected for specific modalities, such as CT and MRI data. Other imaging modalities may provide images with relatively lower variations in the image intensity levels, reflected by lower entropy values and correspondingly lower prediction differences. Moreover, it has been found that specific image types may provide higher or lower characteristic entropy values. In the medical diagnostics field, such image types may include specific anatomies, such as chest, head, extremities, and so forth, providing more or less variation, and stronger or weaker edge lines and contrast. The specific family of compression code tables, then, are preferably established based upon the typical images to be compressed by the system.

Returning to Fig. 6, with the family of compression code tables generated and stored at step 270, configuration segment 250 continues with generation of predictor preferences at indicated at step 272. As described above, rather than employ actual pixel intensity values for encoding in the present compression

technique, difference values may be employed based upon the predictor algorithms. The predictor algorithms generally produce relatively low values in low entropy regions of the image, and relatively higher values in higher entropy regions. However, the use of prediction errors for the data compression will generally result in compression of values which are lower (i.e. shorter in length) than the original data stream code values due to the relatively gradual progression of intensity variations over the image.

In the presently preferred embodiment, one or more of several predictor algorithms may be employed. Referring again to Fig. 5, in a simple and straightforward algorithm, the value of each pixel p(i, j) is predicted to be the value of the immediately preceding pixel p(i-1, j). This predictor algorithm provides a computationally extremely efficient tool for the prediction of each pixel value, with the first pixel in the image being estimated at a value of zero. The difference value is then generated by finding the absolute value of the difference between the predicted value and the actual value for the pixel of interest. The resulting difference values form a matrix of equal size to the original image matrix.

Several alternative predictor algorithms are presently contemplated, and others may be employed as well. In the presently preferred embodiment, these predictor algorithms may be summarized as follows:

$$Ip(i, j) = Ip(i-1, j) \quad \text{(P1)};$$

$$Ip(i, j) = Ip(i-1, j) + Ip(i, j-1) - Ip(i-1, j-1) \quad \text{(P2)};$$

$$Ip(i, j) = ((3*(Ip(i-1, j)) + Ip(i-1, j-1) + Ip(i, j-1) + Ip(i+1, j-1))/6 \quad \text{(P3)};$$

$$Ip(i, j) = ((-2*I(i-1, j-1)) + (3* Ip(i, j-1)) + (3*Ip(i-1, j)))/4 \quad \text{(P4)};$$

$$Ip(i, j) = ((-5*I(i-1, j-1)) + (7* Ip(i, j-1)) +$$

$$Ip(i+1, j-1) + (9*Ip(i-1, j)))/12 \quad \text{(P5)};$$

$$Ip(i, j) = Ip(i, j-1) + Ip(i-1, j))/2 \quad \text{(P6)};$$

where the notation "I" represents the pixel intensity, and the individual pixel designations are mapped in accordance with Fig. 5.

Again, it has been found that various predictors are particularly useful for various types of images, images resulting from various modalities, and various features or anatomies visible in the reconstructed image. For example, the predictor algorithm P1 provides an extremely simple and rapid "last value" estimate of each current pixel value. Predictor P2 provides some degree of edge detection capability, but may accentuate the influence of noise in the image. The third predictor value P3 is a linear predictor shown useful on certain image types and employs values for more of the surrounding pixels, weighting the last value more heavily. In practice, the predictor preferences generated at step 272 of Fig. 6 will follow empirical results for various image types and modalities.

The final step illustrated in segment 250 of the control logic of Fig. 6 is the generation of subregion preferences for the division of the image data stream into subregions. As indicated above, while the lengths of subregions of the image data stream may be different or may vary depending upon such factors as the entropy level, in the presently preferred configuration, the subregion lengths are set equal to one another, but may vary depending upon such factors as the modality originating the image data, the image type, anatomy illustrated in the reconstructed image, and so forth. For example, subregion lengths of 32 pixels are presently set to a default, but alternative subregion lengths of 8, 16, 24, or more than 32 pixels, are contemplated. In general, the preferences set at step 274 in the present embodiment are dependent upon the number of columns in the image matrix, with subregion lengths of 8 being used for matrices having 64 columns or less, subregion lengths of 16 being used for matrices having 128 columns or less, and so forth. To reduce the

computational complexity and to improve speed, in the presently preferred embodiment, the subregions are taken along the major direction or axis (i.e. rows or columns) of the image matrix, depending upon the direction in which the data is stored. Also, the subregions are preferably one dimensional only.

With the compression algorithms, tables, predictor preferences, and subregion size preferences set, the control logic illustrated in Fig. 6 advances through segment 252. In that segment of the logic, the image data is received as indicated at step 276. As noted above, the image data may be directly received from an external source, or may be accessed from a memory within the PACS itself. At step 278, any descriptive information used to select algorithms or parameters of the compression routine are retrieved from the database server. As noted above, such descriptive information may be stored in databases, and may complement the information contained within the descriptive header of the image data stream.

Following segment 252, control advances to evaluation segment 254. Within this segment, the image data is reviewed for descriptive information as indicated step 280. As described above, where descriptive information is available, such as DICOM compliant data in a descriptive header section of the image data stream or descriptive data from the database, some or all of this data is reviewed at step 280. Based upon the preferences set in the configuration segment 272, predictors are selected at step 282 depending upon the image characteristics identified as step 280. Again, these may include the modality of the originating imaging system, the study type or anatomy featured in the image, the number of columns in the image, the number of rows, and so forth. Moreover, other factors may be considered in selecting the predictors at step 282, such as the computational efficiency desired, the processing power of the system, and so forth, with computationally efficient predictors being selected where such processor capabilities are limited, or where additional speed is desired. At step 284, the subregion size for division of the image data stream into subregions is selected in accordance with the preferences established at step 274. Again, step 284 may consist of a default

selection, which may be altered depending upon some or all of the characteristics or factors considered for selection of the predictors.

Evaluation segment 254 continues with the selection of a subset of compression tables, where appropriate, as indicated at step 286. In particular, based upon certain image characteristics, it may be useful to preselect certain compression tables as defaults. For example, specific image types originating in specific modalities, such as CT or MR images, may be best compressed with specific candidate tables which may be selected at step 286. At step 288 a compression header is affixed to the image data. As described below, this compression header will contain code identifying the version of the compression routine, the predictors selected at step 282, the subregion sizes selected at step 284, and so forth.

At compression segment 256 of the control logic, a set of prediction errors or difference values are computed at step 290. As noted above, these values are based upon application of one or more of the predictor algorithms selected at step 282, and by subsequently comparing the predicted values with the actual values for each pixel to determine the prediction error or difference. The resulting prediction errors then form a data stream with the first pixel being assigned its actual value, followed by the difference values for each pixel of the subregion.

The present technique provides for optimal compression of each of the subregions based upon appropriate selection of tables from the family of compression code tables established at step 270. To determine which tables provide best compression, at step 290 of the control logic of Fig. 6, candidate lengths of compressed data are computed by application of each of the candidate compression code tables to the data resulting from application of the predictor algorithms. The total data stream length for each subregion is summed for each case (i.e. application of each of the compression code tables). Following completion of each subregion, the resulting sums are compared to identify which compression code table results in the shortest data stream for the subregion. The corresponding compression code

table is then selected as indicated at step 292. At step 294, code identifying the selected table for each subregion is then inserted into the compressed data and its header, as described in greater detail below.

As noted above, evaluation segment 254 or compression segment 256 may include defaults and constraints on the selection of the optimal code tables. For example, depending upon such factors as the bit depth of the image data to be encoded, certain default selections among the compression code tables may be made, such as selection of the first four tables for bit depths of 8 or less. In addition, certain preferences for designated compression code tables may be made in this selection process, depending upon such factors as the modality originating the image data. By way of example, CT and MRI data may tend to encode higher transitions of pixel intensities, corresponding to higher entropy regions. Accordingly, preferences may be included in the selection of the compression code tables, depending upon the descriptive data reviewed at step 280, such as to prefer one or more higher entropy tables for images originating in such modalities. Finally, in the presently preferred embodiment, code will be inserted directly into the compressed data stream to provide an indication of which of the selected tables is employed for compressing the individual subregions. To optimize the compression, it may be desirable to limit the number of tables which may be selected in the preprocess segment of the control logic to a number which may be conveniently coded in a limited number of bits, such as 2 bits of data. Thus, where table identification codes are to be provided in 2 bits, a constraint may be imposed in the preprocess segment such that only four of the candidate tables may be selected for the compression. Where an additional bit is available for identification of the tables, 8 candidate tables may be selected and encoded.

At step 294 key code for the subregion compression is inserted into the compressed data stream, immediately preceding the compressed subregion data. As noted above, where four candidate tables are employed for the compression, the code inserted at step 294 may include a pair of designated bits. The compression

header created at step 288 cross references this key code to the selected compression tables. At step 296, the image data for the subregion is compressed by application of the selected compression code table. The series of steps of segment 256 is repeated for each subregion of the image until the entire image is compressed.

In final segment 258 the compressed data for each image is completed. Specifically, at step 298, a compression end block is added to the compressed data. At step 300, padding bits are inserted following the compression end block. Finally, at step 302 a checksum value computed through the compression process is added to the compressed data to provide a means for verifying proper decompression. The position and type of code added to the compressed data during segment 258 is described more fully below with reference to Fig. 13.

Where a descriptive header is provided for the original image data, the descriptive header is preferably replaced adjacent to the compression header to complete the image file as indicated at step 260. It will be noted that the resulting data file is a hybrid compressed data file in which the descriptive header data is readable for image management, access, transmission, and similar purposes, while the bulk of the image data is optimally compressed as described above. At step 262, the resulting file is saved, typically in a short term storage system, or in an archive. At step 264 the image is accessed for reviewing, analysis, hard copy creation, and so forth. Upon access of the image, the decompression algorithms identified in the hybrid compressed data file are retrieved for decompression, and the compressed image data is decompressed, as indicated at step 266. This decompression follows the compression header information, with the same compression code tables being employed to decompress the data as were employed to compress it for each subregion. This application of the compression code tables results in identification of the prediction errors or differences on which the compression was based, and the specific predictor algorithm or algorithms employed to generate the differences are then used to regenerate the original image data in a lossless fashion.

Fig. 13 represents an image data set compressed in accordance with the foregoing technique. The image data set, designated generally by the reference numeral 194, includes the descriptive header 196 appended at step 160, along with the compression header 198 affixed at step 192. Following the compression header 198 is the compressed image data 200 which includes compressed image subregions 202, 204 and so forth. Each compressed data subregion, in turn, includes an identification of the algorithm or algorithms (e.g. compression code tables) used to encode the subregion, followed by the actual compressed data.

Fig. 13 also illustrates a presently contemplated format for a compression header 198. As shown in Fig. 13, the compression header includes an identification of the compression routine version 206, followed by an identification of the predictors used in the compression process 207, and identification of the subregion length at reference numeral 208. A compression algorithm identification segment 210 is then inserted in the compression header to identify which of the compression algorithms (e.g. compression code tables) were selected for the compression, and the manner in which each table is encoded in the data stream for each subregion. In the example of Fig. 13, for example, tables 0, 3, 4 and 5 were selected, as indicated at reference numeral 212, with each table being identified by a 2 bit binary code to be found within the first 2 bits of the compressed data for each subregion.

Fig. 13 also illustrates the format in the presently preferred embodiment for the compressed data code for each image subregion 202, 204, and so forth. In the illustrated embodiment, a first portion of each subregion includes an identification of the compression table used to compress the subregion data, as indicated at reference numeral 214 for subregion 202, and reference numeral 218 for subregion 204. This key code is followed by the compression codes as indicated at reference numerals 216 and 220, respectively. Finally, Fig. 13 illustrates the code inserted at the end of each compressed image file, or portion of a file. Specifically, following the last subregion BN of the image, the end of block code portion 222 is affixed. As noted above, this end of block code signals the end of an image, or may be used to signal

the end of a portion of an image, where the data compression routine is changed within a single image, such as due to large variations in the image data entropy. The padding code is inserted as indicated at reference numeral 224. This code may be of variable size as needed to complete the compressed data file on a whole word length. Finally, a 32 bit checksum portion 226 is added to complete the compressed data.

The foregoing aspects of the present technique may be adapted in various manners depending upon the type of image to be compressed. For example, the technique may be used on both images composed of varying gray levels, as well as on color images. As will be appreciated by those skilled in the art, color images will typically consist of various color components, which produce the appearance of color variations due to their respective intensities. The foregoing technique may be used either with or without separating the color components from one another, but is preferably applied by separating the color components and processing (i.e. compressing) the components in groups. Similarly, multiframe images may be accommodated in the present technique. As will be appreciated by those skilled in the art, such images typically comprise a number of separate images encoded by rows and columns, without separate descriptive headers (e.g. DICOM compliant headers) positioned between the separate images. In such cases, code identifying offsets in the compressed data corresponding to locations of different image frames is preferably inserted into the descriptive header of the hybrid compressed data file.

It has been found that the foregoing image data compression and decompression technique can be further refined to provide for multiresolution (or multi-size) image data compression to further enhance rates of data transfer and decompression. Where a user does not desire to view a full image with maximum resolution, or where the user view port is limited, such multiresolution image compression facilitates transfer of a reduced size image to the user for viewing, with excellent image quality. Moreover, as described below, the multiresolution image compression aspect of the present technique allows a user to view a reduced size or

reduced resolution image relatively rapidly, and to "zoom" on the image thereafter by transfer of only a portion of the compressed data corresponding to components of the greater sized image not already transferred. The additional data is then processed and combined with the reduced size image data to obtain the larger sized image.

The present multiresolution implementation is based upon lossless integer wavelet decomposition in combination with optimized Huffman code compression as described above, and modification of the Huffman code compression based on a recognition of the nature of high frequency data sets from the wavelet decomposition. Specifically, as will be recognized by those skilled in the art, wavelet decomposition involves a dyadic filtering and sub-sampling process. This creates a hierarchical set of sub-bands, as illustrated in Fig. 14. As illustrated in Fig. 14, an image data set 300 includes low frequency components 302 along with high frequency components 304, which may be considered as noise or variations from the low frequency components. A single level wavelet decomposition results in a decomposed data set 306 which includes one low frequency sub-band LL, designated by reference numeral 308 in Fig. 14, along with three high frequency ones LH, HL, and HH, designated by reference numerals 310, 312 and 314. Subsequent decomposition may be considered to produce a further data set 316 in which the low frequency sub-band is further decomposed into a set 318 of sub-bands, including a low frequency band 320, along with three additional high frequency sub-bands 322, 324 and 326.

Typically wavelet transforms are real (floating point) filters with the result also being real values. Until recently only the S-Transform (a modified Haar wavelet) provided an integer-based transformation and reconstruction. As evident from the nature of the transformation it is very difficult to preserve precision with floating point operations. However, by a technique referred to as "lifting," any wavelet transformation may be implemented as an integer-based transformation with full reversibility.

Using this "lifting" technique various wavelet transforms were analyzed and compared, including the S-Transform (Haar wavelet with lifting), a (2+2,2) transform, and a (4,2) transform. Moreover, analysis was done based on the information theoretic entropy values of the resulting sub-bands. Entropy values in base 2 provide the lower bounds on the average length of the variable length code (VLC) through Kraft's Inequality. In addition to the entropy values, the operations required to represent the values of a single entry in each of the four sub-bands were determined. The outcome of the analysis suggested that the most optimum wavelet to use for the transformation is the S-Transform. Although other transforms may be employed, in the present embodiment, the S-Transform is preferred; with the emphasis being placed on the speed and complexity of the transformation more so than the resulting compression ratios.

For decomposition of the image data set, a forward one step wavelet transform in one dimension is based on the following equations:

$$L(n) = \lfloor (C(2n)+C(2n+1))/2 \rfloor \qquad \text{for } n \in [0,N/2-1]$$

and

$$H(n) = C(2n) - C(2n+1);$$

where $C(i)$ for $i \in [0,N-1]$ represents the input data, L and H are the decomposed low and high frequency components, and C is the input data. The operation designed by the symbols "$\lfloor\ \rfloor$" produces the greatest integer less than the operands with "N" being the size of the input data.

The converse of this transformation, the one step inverse wavelet transform is described by the following equations:

$$C(2n) = L(n) + \lfloor (H(n) + 1)/2 \rfloor$$

and

$C(2n+1) = C(2n) - H(n)$.

An implicit assumption in the foregoing equations is the fact that the data size "N" is even. Though valid for theoretical analysis and description, this assumption may not be valid for certain data sets to be compressed in the present technique. The technique has, however, been adapted to accommodate odd and even sizes of the input data, and to extend the one-dimensional transform to a two-dimensional one, as summarized below.

The equations for the forward and inverse wavelet transforms described above provide one-dimensional single-step transformation. Recursive single-step wavelet decompositions in two dimensions provide the tessellation of the image as depicted in Fig. 14. In the present technique, a recursion of a single-step wavelet transform is performed on the low frequency or "LL" component at every level. The number of levels for the transformation is determined by fixing the row and/or column size of the smallest resolution. This level value is determined by the steps necessary to decompose the maximum of the row or column size of the original image to the desired smallest resolution size. If "n" is this level variable then the following equation is used:

$$n = \log_2(\max(\text{rows},\text{cols})) - \log_2(d_{size})$$

where n is the number of levels of decomposition, "rows" and "cols" are the original image dimensions, $\log_2$ is the log in base 2, and d_{size} is the configurable size of the smallest resolution image.

Special handling of the odd row or column at every level is performed. In a present implementation, the odd row or the odd column is replicated with the aim to force it to be even such that the algorithm for the wavelet transformation is a seamless unit. While this addition adds somewhat to the memory space needed for image storage, the additions are negligible when compression is performed due

to the fact that the high frequency sub-bands will have all zeros in those rows or columns.

Referring to the nomenclature employed above, the two-dimensional one-step forward transformation to decompose a quad o input image data (a, b, c, d) to *ll*, *hl*, *lh*, *hh* is governed by the following equations:

$$ll = \lfloor (\lfloor (a+b)/2 \rfloor + \lfloor (c+d)/2 \rfloor)/2 \rfloor;$$

$$hl = \lfloor ((a-b) + (c-d))/2 \rfloor;$$

$$lh = \lfloor (a+b)/2 \rfloor - \lfloor (c+d)/2 \rfloor; \text{ and}$$

$$hh = (a-b) - (c-d).$$

In addition to handling the odd row and odd column numbers at every level of the transform, the present technique has been adapted to accommodate the possibility of overflow when using signed or unsigned 16 bit arrays. While all values could be treated as 32 bit values, loss of speed in performing the desired transformations may result, as well as a significant increase in the memory required to process the image. In a present implementation designed for PACS, because a significant majority of the data values lie within 14 bits for unsigned and signed data, to preserve the speed of the algorithm and to create the minimum possible memory map, 16 bit signed and unsigned routines are employed as a default, while checking for overflow conditions, as described below. When an overflow condition is encountered, the 16 bit-based transform routine exits to the appropriate overflow routine.

Fig. 15 illustrates diagrammatically the progressive processing of image data by wavelet decomposition and Huffman code compression to produce a data stream for storage, access and transmission. The general process flow, designated

generally by the reference numeral 328 in Fig. 15, begins with the image data set 330, which may be acquired in any suitable manner and which may be preprocessed, such as for dynamic range adjustment, image enhancement, and so forth. This image, designated as a level 0 in the example, is then decomposed by forward wavelet transformation as described above, to provide a data set 332, itself comprising LL, LH, HL and HH data sets. The low frequency data set may then be further decomposed by similar forward wavelet transformation to generate a next level data set 334. Further wavelet transformation on the low frequency data set at each successive level provides a similar data set, until a final data set 336 is reached, which is considered the smallest or lowest resolution data set desired, denoted "n" in the present example.

The data sets obtained by successive wavelet decomposition are then compressed as follows. At each level, the high frequency data sets are compressed in accordance with a modified compression routine, as designated by reference numeral 338. In this modified routine, all processing is substantially identical to that described above, with the exception that the actual values of the data sets are used rather than predicted error values. That is, steps 280-290 illustrated in Fig. 6 are not performed, while other steps, such as subregion analysis, optimum code table selection, and so forth, are performed in substantially the same manner. The modified compression routine performed on the high frequency wavelet transformed data sets is based upon a realization that the wavelet transformed data is already decorrelated in a manner that enables the present compression technique to be applied directly without the need to predict or compute error values.

Because the low frequency data sets for each higher level is further decomposed, information descriptive of these data sets is preserved in the lower levels, with the exception of the lower-most low frequency data set (i.e., the low frequency data set at the nth level). In the present implementation, this data set is compressed in accordance with the predictive error compression technique described above, as indicated by reference numeral 340 in Fig. 15.

Several points should be noted regarding the decomposition and compression summarized in Fig. 15. First, as will be appreciated by those skilled in the art, depending upon the size and type of data analyzed, the values in the Huffman code tables described above, the subregion lengths, and other specific parameters may be adapted to provide the desired compression ratios and speed of compression. Moreover, because successive decomposition of the low frequency data sets results in progressively smaller data sets (halved in both dimensions), these data sets offer the possibility of access, transmission and reconstruction at any of the available levels for the multiresolution compression and decompression aspect of the present technique.

Following compression of the high and low frequency data sets as summarized in Fig. 15, the resulting data is compiled in a data stream or file as indicated by reference numeral 342. In a present embodiment, the data stream includes a descriptive header 344, followed by a series of data series, including a first series 346 for the lower-most resolution data set (including the low and high frequency compressed data), and a successive data sets 348, each including high frequency compressed data for the respective level. In a present implementation, the data stream or file includes a header describing the version of the multiresolution scheme, the type of forward transform (sign/unsigned with/without overflow, see the discussion below), the number of levels of wavelet decomposition, the row and column values of every sub-band level (resolution), and the compressed sizes of all the sub-bands from smallest to the largest. Again, the lowest level low frequency data set is compressed using predictive compression as described above, while the high frequency data sets for each level are compressed using the appropriate non-predictive or modified compression scheme. In addition to storing the data storage header at the top of the compressed stream, in a medical diagnostic context, other elements may be stored in a DICOM header. In a present implementation, these include the number of levels "n," row (rr) values (rr(n), rr(n-1),..., rr(0)), column (cc) values (cc(n), cc(n-1),..., cc(0)),

and compressed data size (cxebyte) for every level (cxebyte(n), cxebyte(n-1),..., cxebyte(0)).

As noted above, in order to provide an efficient implementation from processing and storage standpoints, a present embodiment provides for 16 bit processing, with 32 bit processing only where required by the input data. Referring to Fig. 16, logic for wavelet transformation in accordance with this approach is summarized, as indicated by reference numeral 350. In this implementation, four forward transform routines were coded, one each for signed and unsigned 16 bit data and one each to handle the signed and unsigned overflow conditions (the overflow routines were 32 bit versions of their 16 bit counterparts). Thus, at step 352, a forward transform is performed on the signed and unsigned data. At step 354, the data is analyzed to determine whether overflow processing is needed. If not, the transform coefficients are returned, as noted at step 356, and processing continues as described above. If, however, 32 bit processing is needed, the logic diverts to step 358, where the transform is restarted with overflow, and the transform coefficients from such processing are then returned as noted at step 360. Similar techniques may apply for the processing of 8 bit and color data types.

When combined with the compression used at the lowest or nth level of the wavelet decomposed data sets, the foregoing scheme may be summarized as illustrated in Fig. 17. The logic, referred to generally by reference numeral 362, begins at step 364, wherein the image data is input for forward transformation. Four routines were coded in a present implementation, corresponding to the four forward routines. At the reconstruction phase, described below, the encoded type flag signals the appropriate inverse transform routine to be used. This simplifies the reconstruction phase. The process flow summarized in Fig. 17 reflects the implementation in which, along with the non-predictive or modified compression scheme, processing has been adapted to handle the 32 bit signed and unsigned data that was generated by the respective overflow forward transformation routines. Thus, at step 366, it is determined whether overflow has occurred requiring 32 bit

processing. If not, the sub-bands or data sets for the successive levels are generated at step 368, and the modified compression technique is carried out for all high frequency data sets of sub-bands, as indicated at step 370. As noted above, the lowest or nth level low frequency data set is compressed in accordance with the predictive compression technique. If, at step 366, it is determined that overflow processing is needed, the sub-bands or data sets are generated at step 374 and the high frequency data sets are compressed via 32 bit processing, as noted at step 376. Again, the low frequency data set for the nth level is compressed at step 372 in accordance with the predictive compression scenario. As noted above, the compression Huffman code tables used for the compression may be adapted in accordance with the exigencies of the data, such at to handle the resulting 18 bit signed and unsigned data. In a present implementation, additional code tables are employed for coding and decoding when the input exceeds 16 bits of storage. The overflow condition has a maximum bit depth of 18 bits signed.

A present implementation of the foregoing technique provides function calls to access all the values that are enumerated above. These elements are not required for the decompression and inverse transform routines as they are present in the header of the compressed stream. However, they are desirable where the user may wish to display images at different resolution levels. The decompression and reconstruction routines may be composed such that for all users no changes to the interface are needed, apart from buffering the reconstructed data. For users with specific needs to display at various resolution levels, user-specific routines may be developed.

A data stream or file map created by the foregoing technique is illustrated in somewhat greater detail in Fig. 18. As discussed above, the data stream or file 342 includes header 344, followed by code valves for the compressed sub-bands or data sets, beginning with the nth data set. Within these data segments, sections 378 encode the low and high frequency sub-bands or data sets. In the following data segments 348, sections 380 encode high frequency sub-bands or data sets for

the higher levels. It should also be noted that, as mentioned above, additional code valves are included within the data stream for the information discussed above with respect to Huffman code tables used for compression, subregion lengths, and so forth.

Fig. 19 summarizes diagrammatically the process for decompressing and reconstructing images from the compressed data file. The process, designated generally by reference numeral 382, includes accessing the data for the lowest level data sets and for any further data sets through the size or resolution to be provided to the user. Thus, as summarized in Fig. 19, the data from the file is accessed, in accordance with the information stored in the header, and decompressed, with the low frequency data set for the lowest level being decompressed, as indicated at reference numeral 384, in accordance with the predictive error technique discussed above. The high frequency sub-bands or data sets are decompressed, as indicated at reference numeral 386, in accordance with the modified compression technique. That is, rather than treating the decompressed data as predicted errors, the resulting data is treated as wavelet transform coefficients directly.

Following decompression of data up to the desired level, inverse wavelet transformation is performed on the decompressed data, with the lower level image data sets serving as the low frequency sub-band for the next higher image level. Thus, the inverse transform routine operates by treating the smallest resolution "LL" band and combining it with its associated "HL", "LH", and "HH" bands to produce the next higher resolution "LL" band. This process is repeated until either the full resolution of the image is achieved or a specified level of resolution is attained. The one step inverse transform takes the ll, hl, lh, and hh values, that is, the values for the lowest level, and produces the quad pixels for the next level ll band (at the final inverse transformation the full resolution pixel data for the image is produced).

Again, referring to the nomenclature employed above, the one-step two-dimensional inverse transform is governed by the following set of equations:

$$a = ll + \lfloor (hl+1)/2 \rfloor + \lfloor (lh + \lfloor (hh+1)/2 \rfloor) + 1)/2 \rfloor;$$

$$b = ll + \lfloor (hl+1)/2 \rfloor + \lfloor ((lh + \lfloor (hh+1)/2 \rfloor) + 1)/2 \rfloor - (lh + \lfloor (hh+1)/2 \rfloor);$$

$$c = ll + \lfloor (hl+1)/2 \rfloor - hl + \lfloor (lh + \lfloor (hh+1)/2 \rfloor - hh + 1)/2 \rfloor; \text{ and}$$

$$d = (ll + \lfloor (hl+1)/2 \rfloor - hl + \lfloor (lh + \lfloor (hh+1)/2 \rfloor - hh + 1)/2 \rfloor) - ((lh + \lfloor (hh+1)/2 \rfloor) - hh).$$

In a present implementation, the design for the inverse transform was made modular with respect to possible single level reconstruction. This allows users to specify a desired level, from the smallest resolution to full resolution, for reconstruction. Moreover, the encoding of the type of forward transform in the compressed data stream allows for the reconstruction using the correct inverse transform routine.

It should be noted that the foregoing technique permits access, transmission and reconstruction of images at any one of the stored sizes and resolutions. Moreover, where zooming or enhanced resolution is desired from an image already accessed by a user, only the additional high frequency bands for the higher resolution level need be accessed and processed for combination with the data already provided for reconstruction of the new image.

The multiresolution aspect of the technique thus provides targeted decompression and reconstruction for a specific level image between 0 and "n." The user provides the appropriate subset of the compressed stream from information present in the header, such as a DICOM header in a medical diagnostic application, with indication of the desired level of reconstruction to a

level selection routine. A pixel buffer then allows the user to store the expanded and reconstructed image.

The foregoing novel functionality allows the user to efficiently use available transmission bandwidth. Exemplary compression results are summarized in table 388 of Fig. 20 for an image produced in a medical diagnostic context. In the example, a typical chest CR with image row size of 2048 and column size of 2500 is transformed to five levels. The relevant information of row and column sizes per level and the associated compressed bytes are shown. The 10MB image is multiresolution compressed to a size of 3,909,848 bytes (the sum of the numbers of bytes listed for each image level in the table). Thus, to reconstruct the full resolution 2048x2500 image would require the transmission of approximately 3.9MB of data.

The user may, however, access much smaller amounts of data as follows. By way of example, it is assumed that in a typical web-client application the resolution size of a user monitor is about 1k x 1k. Reviewing the data set forth in table 388, this size limitation would approximately permit reconstruction to level 1. Thus, data required for transmission from the server equals 979,988 bytes (the sum of the bytes comprising the images through level 1. However, in many cases a subset of the full monitor resolution is allocated for the image display component. Consequently, in the example, a first image may be reconstructed to level 2 in a very rapid process, particularly as compared to transmission and reconstruction of the highest resolution image, requiring only 255,428 bytes to be transmitted. As the multi-resolution scheme allows for subsequent zooming to higher levels, then, the remaining 724,560 bytes to reconstruct level 1 from the level 2 image, and subsequently the remaining 2,929,860 bytes to reconstruct the full resolution image from a level 1 reconstructed image.

While the invention may be susceptible to various modifications and alternative forms, specific embodiments have been shown by way of example in

the drawings and have been described in detail herein. However, it should be understood that the invention is not intended to be limited to the particular forms disclosed. Rather, the invention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the invention as defined by the following appended claims.

Substitute Sheet

CLAIMS:

1. A method for compressing image data, the method comprising the steps of:

decomposing image data into a plurality of data sets for different resolutions;

compressing the data sets; and

storing the compressed data sets in a data stream that includes groups of compressed data sets for each of the different resolutions.

DRAFT - 2015-08-25

ABSTRACT OF THE DISCLOSURE

Image data is compressed by decomposing the image data into a plurality of resolutions or sized by wavelet decomposition, followed by compression of the resulting data sets. Adaptive Huffman code compression is optimized for compressing the data sets, predictive errors for a low frequency band and the wavelet transformed values for high frequency bands. The image data is thus compressed losslessly and in multiple resolutions such that desired resolution images can be reconstructed and transmitted in accordance with available bandwidth and viewing capabilities.

Express Mail Label No. EL 929 606 353 US

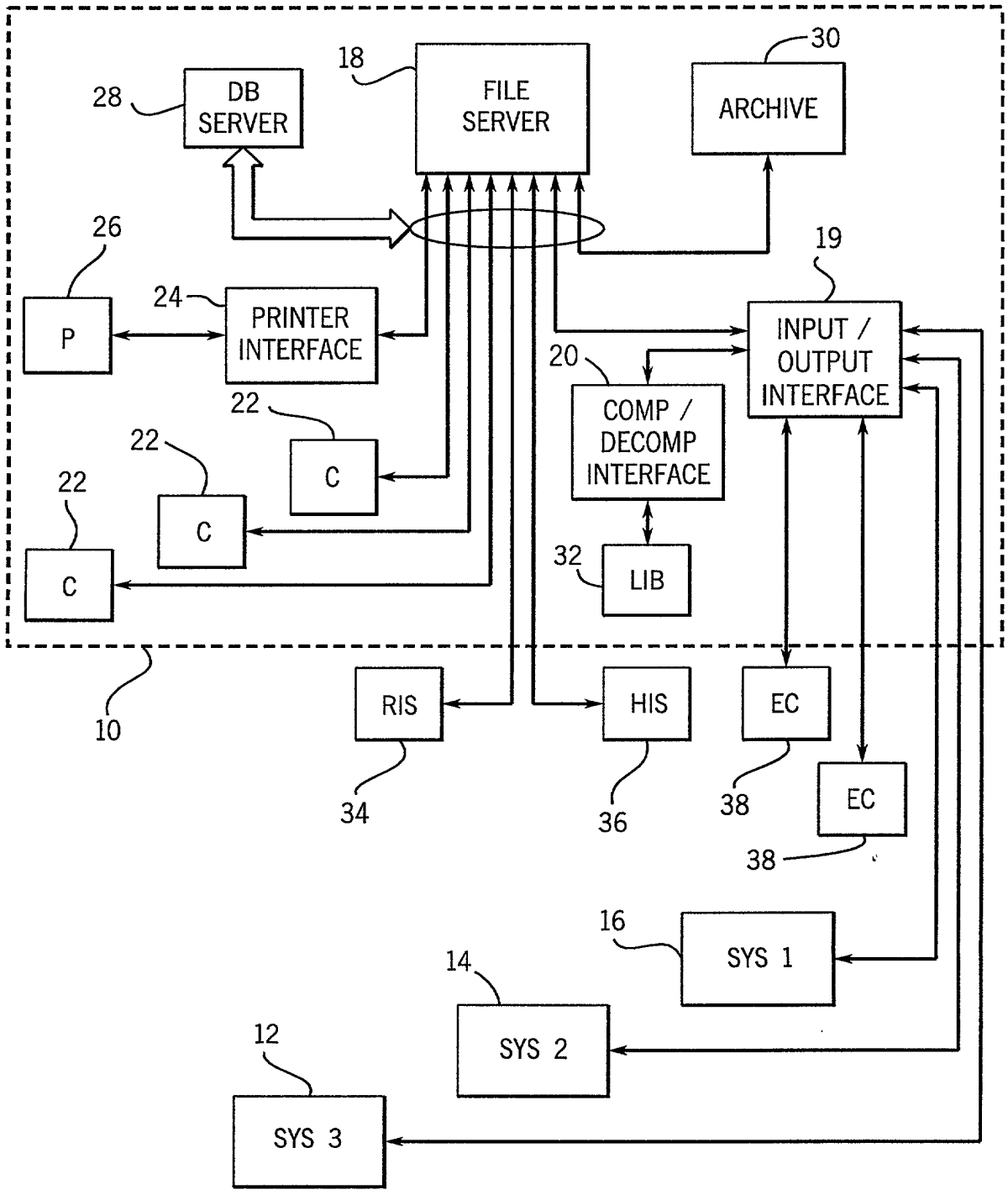

18
FILE SERVER
28
DB SERVER
30
ARCHIVE
26
P
24
PRINTER INTERFACE
19
INPUT / OUTPUT INTERFACE
20
COMP / DECOMP INTERFACE
22
C
22
C
22
C
32
LIB
10
RIS
HIS
EC
34
36
38
EC
38
16
SYS 1
14
SYS 2
12
SYS 3

FIG. 1

FIG. 2

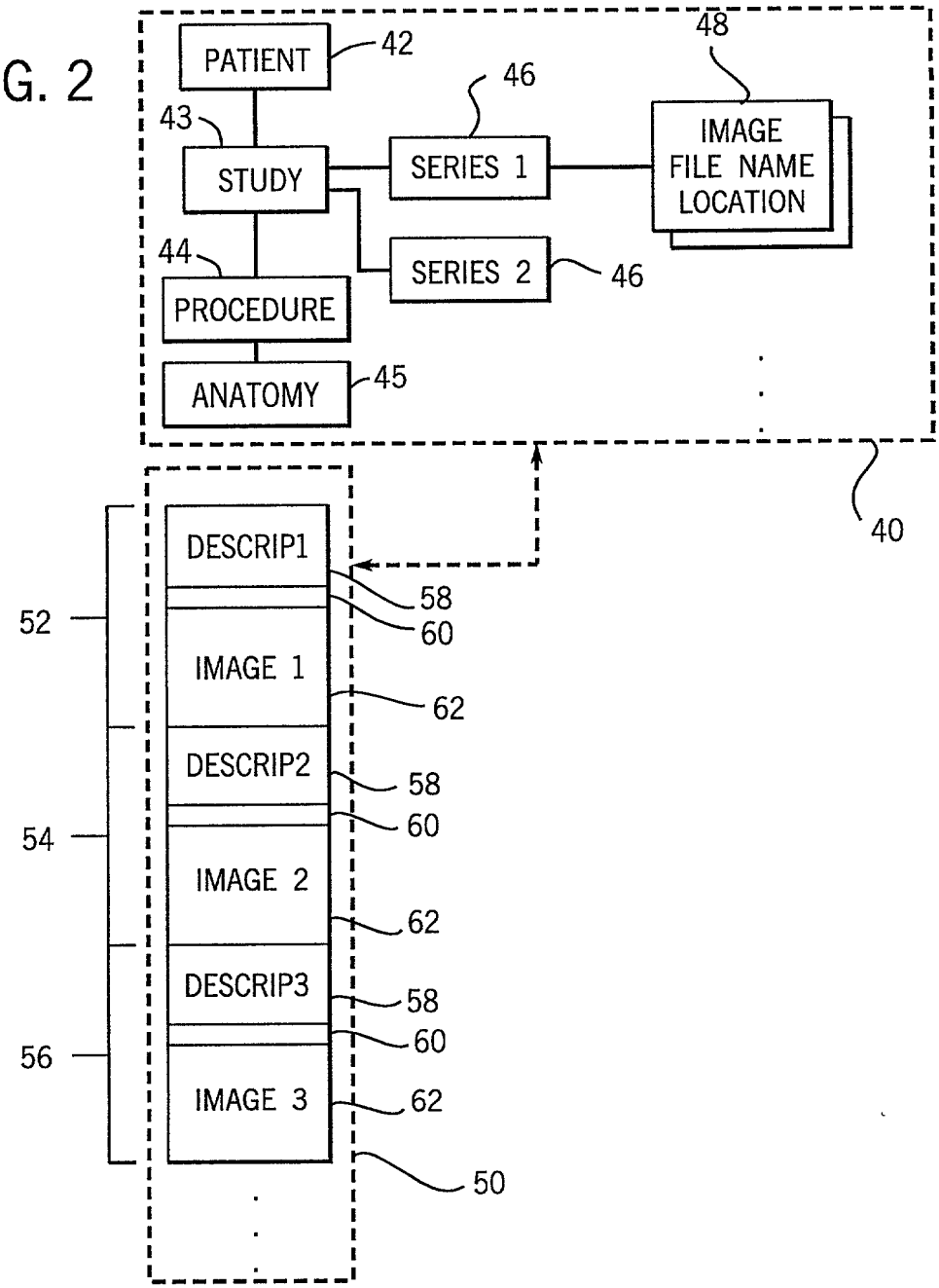
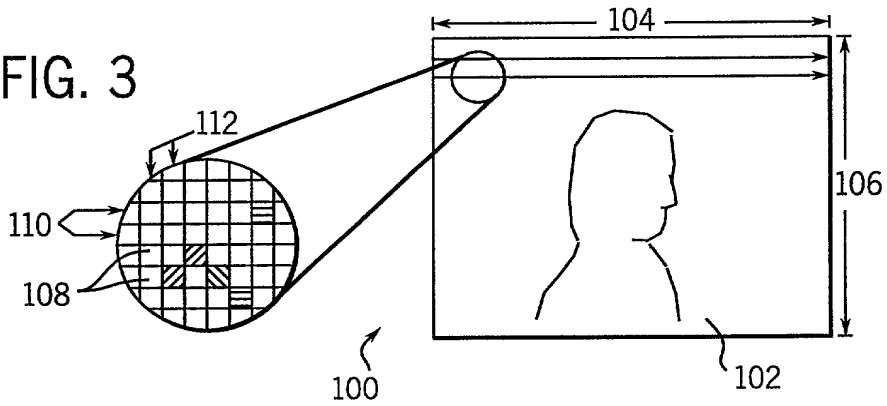

PATIENT
42
48
46
43
STUDY
SERIES 1
IMAGE
FILE NAME
LOCATION
44
SERIES 2
46
PROCEDURE
ANATOMY
45
40
DESCRIP1
58
60
52
IMAGE 1
62
DESCRIP2
58
60
54
IMAGE 2
62
DESCRIP3
58
60
56
IMAGE 3
62
50

FIG. 3


104
112
110
106
108
100
102

Docket: 5960-3/5960

FIG. 4

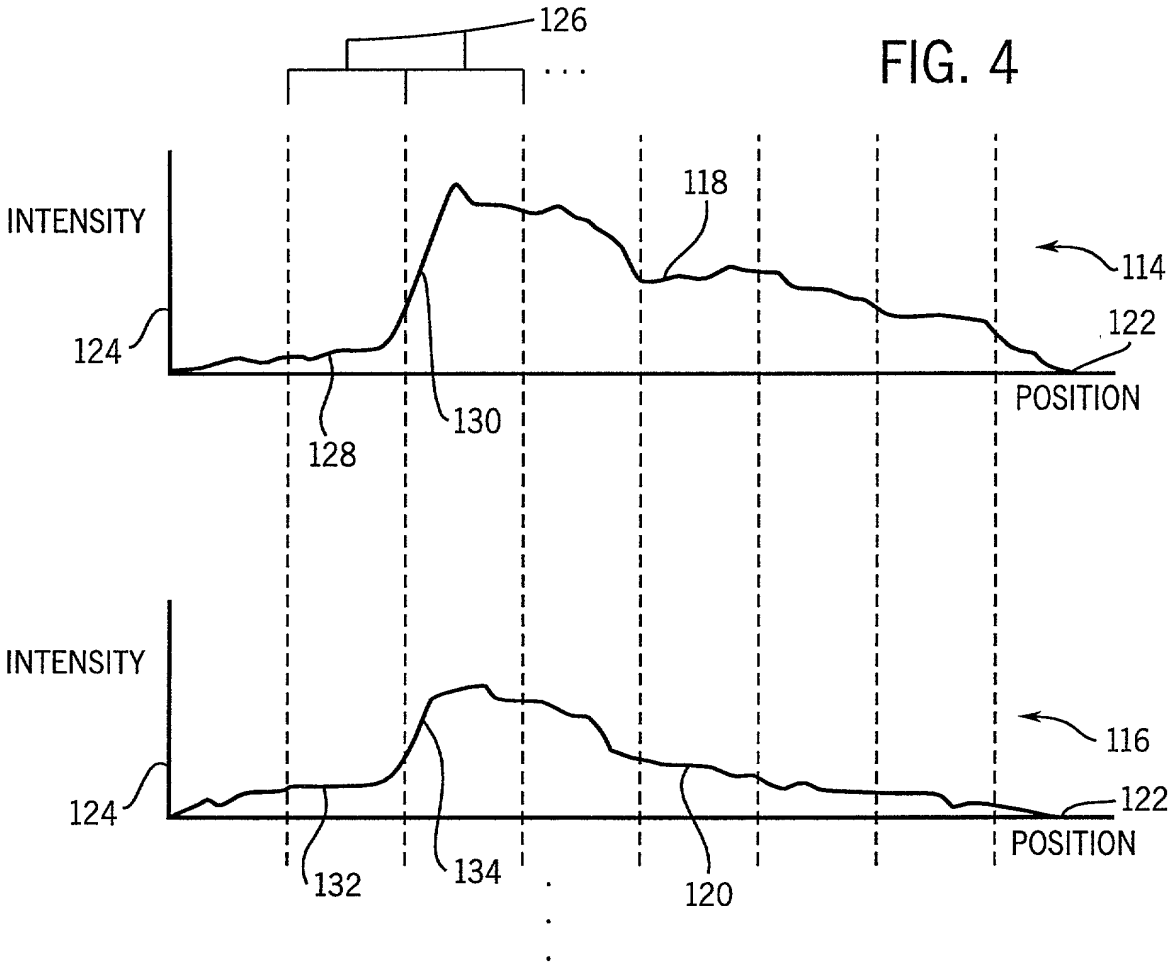
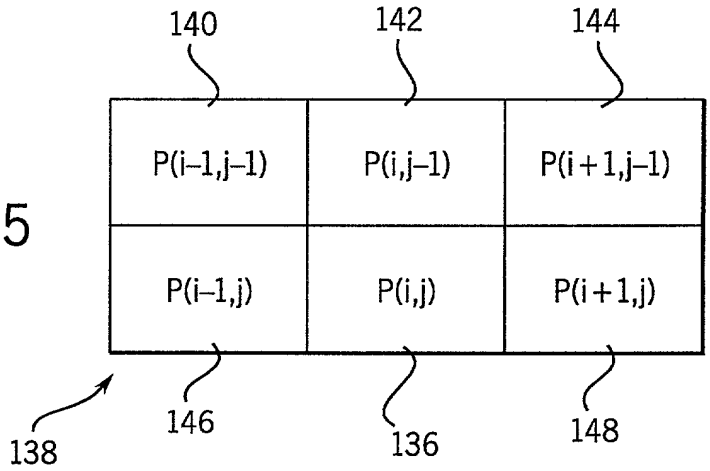


FIG. 5



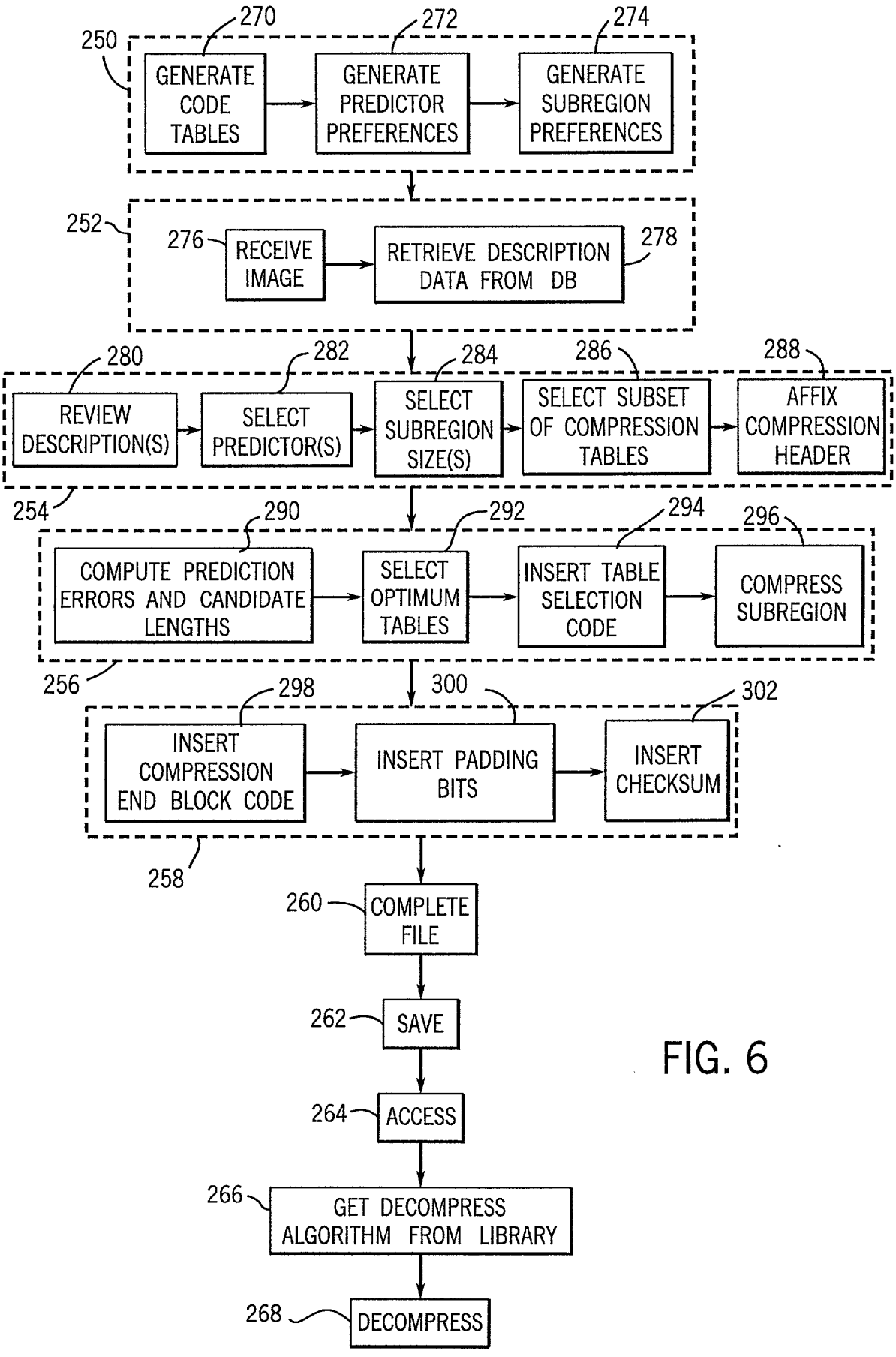

250
270
GENERATE CODE TABLES
272
GENERATE PREDICTOR PREFERENCES
274
GENERATE SUBREGION PREFERENCES
252
276
RECEIVE IMAGE
RETRIEVE DESCRIPTION DATA FROM DB
278
254
280
REVIEW DESCRIPTION(S)
282
SELECT PREDICTOR(S)
284
SELECT SUBREGION SIZE(S)
286
SELECT SUBSET OF COMPRESSION TABLES
288
AFFIX COMPRESSION HEADER
256
290
COMPUTE PREDICTION ERRORS AND CANDIDATE LENGTHS
292
SELECT OPTIMUM TABLES
294
INSERT TABLE SELECTION CODE
296
COMPRESS SUBREGION
258
298
INSERT COMPRESSION END BLOCK CODE
300
INSERT PADDING BITS
302
INSERT CHECKSUM
260
COMPLETE FILE
262
SAVE
264
ACCESS
266
GET DECOMPRESS ALGORITHM FROM LIBRARY
268
DECOMPRESS

FIG. 6

FIG. 7

16 – BIT LOSSLESS ENCODING
SCHEME 0 – MAX RATIO = 16:1

HUFFMAN CODE PREFIX	CODE EXTENSION	DIFFERENCE d	CODE LENGTH	ENCODED RANGE WIDTH
0		0	1	0
100		+1	3	+1
101		–1	3	
1100	X	+2 TO +3	5	+2
1101	X	–2 TO –3	5	
11100	XX	+4 TO +7	7	+4
11101	XX	–4 TO –7	7	
111100	XXX	+8 TO +15	9	+8
111101	XXX	–8 TO –15	9	
1111100	XXXX	+16 TO +31	11	+16
1111101	XXXX	–16 TO –31	11	
11111100	XXXXX	+32 TO +63	13	+32
11111101	XXXXX	–32 TO –61	13	
111111100	XXXXXX	+64 TO +127	15	+64
111111101	XXXXXX	–64 TO –127	15	
1111111100	XXXXXXX	+128 TO +255	17	+128
1111111101	XXXXXXX	–128 TO –255	17	
11111111100	XXXXXXXX	+256 TO +511	19	+256
11111111101	XXXXXXXX	–256 TO –511	19	
11111111110<16 BITS)		ACTUAL VALUE	27	
11111111111		END OF BLOCK	11	

180 172 182 174 176 170 178

FIG. 8

SCHEME 1 – MAX RATIO = 5.33:1

HUFFMAN CODE PREFIX	CODE EXTENSION	DIFFERENCE d	CODE LENGTH	ENCODED RANGE WIDTH
00	X	0 TO +1	3	1
01	X	–1 TO –2	3	
100	XX	+2 TO +5	5	+4
101	XX	–3 TO –6	5	
1100	XXX	+6 TO +13	7	+8
1101	XXX	–7 TO –14	7	
11100	XXXX	+14 TO +29	9	+16
11101	XXXX	–15 TO –30	9	
111100	XXXXX	+30 TO +61	11	+32
111101	XXXXX	–31 TO –62	11	
1111100	XXXXXX	+62 TO +125	13	+64
1111101	XXXXXX	–63 TO –126	13	
11111100	XXXXXXX	+126 TO +253	15	+128
11111101	XXXXXXX	–127 TO –254	15	
111111100	XXXXXXXX	+254 TO +509	17	+256
111111101	XXXXXXXX	–255 TO –510	17	
1111111100	XXXXXXXXX	+510 TO +1021	19	+512
1111111101	XXXXXXXXX	–511 TO –1022	19	
1111111110<16 BITS)		ACTUAL VALUE	26	
1111111111		END OF BLOCK	10	

184

US 6,396,948 B1

FIG. 9

SCHEME 2 – MAX RATIO = 4:1

HUFFMAN CODE PREFIX	CODE EXTENSION	DIFFERENCE d	CODE LENGTH	ENCODED RANGE WIDTH
00	XX	0 TO +3	4	3 ←186
01	XX	–1 TO –4	4	
100	XXX	+4 TO +11	6	+8
101	XXX	–5 TO –12	6	
1100	XXXX	+12 TO +27	8	+16
1101	XXXX	–13 TO –28	8	
11100	XXXXX	+28 TO +59	10	+32
11101	XXXXX	–29 TO –60	10	
111100	XXXXXX	+60 TO +123	12	+64
111101	XXXXXX	–61 TO –124	12	
1111100	XXXXXXX	+124 TO +251	14	+128
1111101	XXXXXXX	–125 TO –252	14	
11111100	XXXXXXXX	+252 TO +507	16	+256
11111101	XXXXXXXX	–253 TO –508	16	
111111100	XXXXXXXXX	+508 TO +1019	18	+512
111111101	XXXXXXXXX	–509 TO –1020	18	
111111110 <16 BITS)		ACTUAL VALUE	25	
111111111		END OF BLOCK	9	

FIG. 10

SCHEME 3 – MAX RATIO = 3.2:1

HUFFMAN CODE PREFIX	CODE EXTENSION	DIFFERENCE d	CODE LENGTH	ENCODED RANGE WIDTH
00	XXX	0 TO +7	5	7 ←188
01	XXX	–1 TO –8	5	
100	XXXX	+8 TO +23	7	+16
101	XXXX	–9 TO –24	7	
1100	XXXXX	+24 TO +55	9	+32
1101	XXXXX	–25 TO –56	9	
11100	XXXXXX	+56 TO +119	11	+64
11101	XXXXXX	–57 TO –120	11	
111100	XXXXXXX	+120 TO +247	13	+128
111101	XXXXXXX	–121 TO –248	13	
1111100	XXXXXXXX	+248 TO +503	15	+256
1111101	XXXXXXXX	–249 TO –504	15	
11111100	XXXXXXXXX	+504 TO +1015	17	+512
11111101	XXXXXXXXX	–505 TO –1016	17	
11111110 <16 BITS)		ACTUAL VALUE	24	
11111111		END OF BLOCK	8	

Document: 2/6/2004 11:59:09 AM

FIG. 11

SCHEME 4 – MAX RATIO = 2.67:1

HUFFMAN CODE PREFIX	CODE EXTENSION	DIFFERENCE d	CODE LENGTH	ENCODED RANGE WIDTH
00	XXXX	0 TO +15	6	15
01	XXXX	–1 TO –16	6	
100	XXXXX	+16 TO +47	8	+32
101	XXXXX	–17 TO –48	8	
1100	XXXXXX	+48 TO +111	10	+64
1101	XXXXXX	–49 TO –112	10	
11100	XXXXXXX	+112 TO +239	12	+128
11101	XXXXXXX	–113 TO –240	12	
111100	XXXXXXXX	+240 TO +495	14	+256
111101	XXXXXXXX	–241 TO –496	14	
1111100	XXXXXXXXX	+496 TO +1007	16	+512
1111101	XXXXXXXXX	–497 TO –1008	16	
1111110 <16 BITS)		ACTUAL VALUE	23	
1111111		END OF BLOCK	7	

190

FIG. 12

SCHEME 5 – MAX RATIO = 2.29:1

HUFFMAN CODE PREFIX	CODE EXTENSION	DIFFERENCE d	CODE LENGTH	ENCODED RANGE WIDTH
00	XXXXX	0 TO +31	7	31
01	XXXXX	–1 TO –32	7	
100	XXXXXX	+32 TO +95	9	+64
101	XXXXXX	–33 TO –96	9	
1100	XXXXXXX	+96 TO +223	11	+128
1101	XXXXXXX	–97 TO –224	11	
11100	XXXXXXXX	+224 TO +479	13	+256
11101	XXXXXXXX	–225 TO –480	13	
111100	XXXXXXXX	+480 TO +991	15	+512
111101	XXXXXXXX	–481 TO –992	15	
111110 <16 BITS)		ACTUAL VALUE	22	
111111		END OF BLOCK	6	

192

US 5,956,674

0002TT "E09ST260

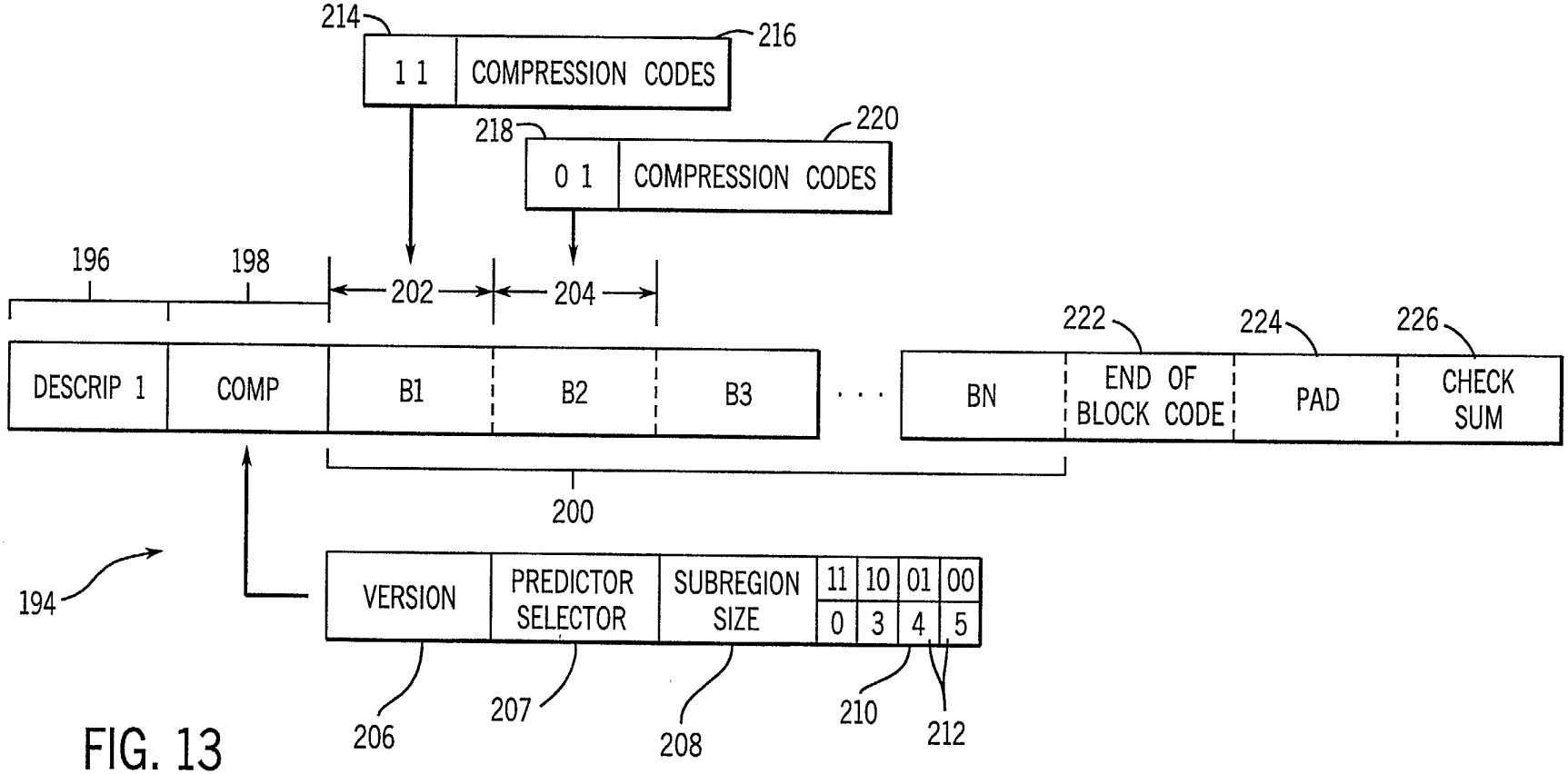


FIG. 13

FIG. 14

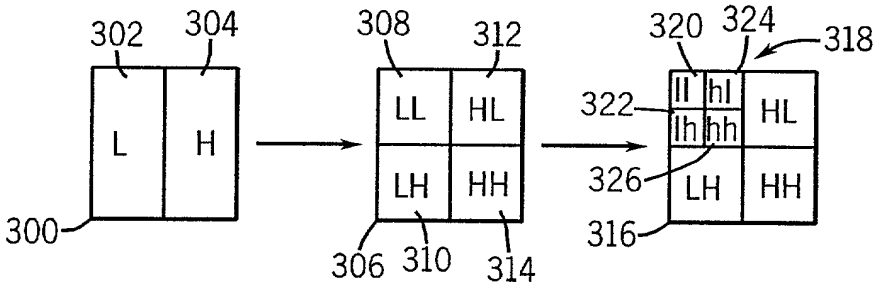
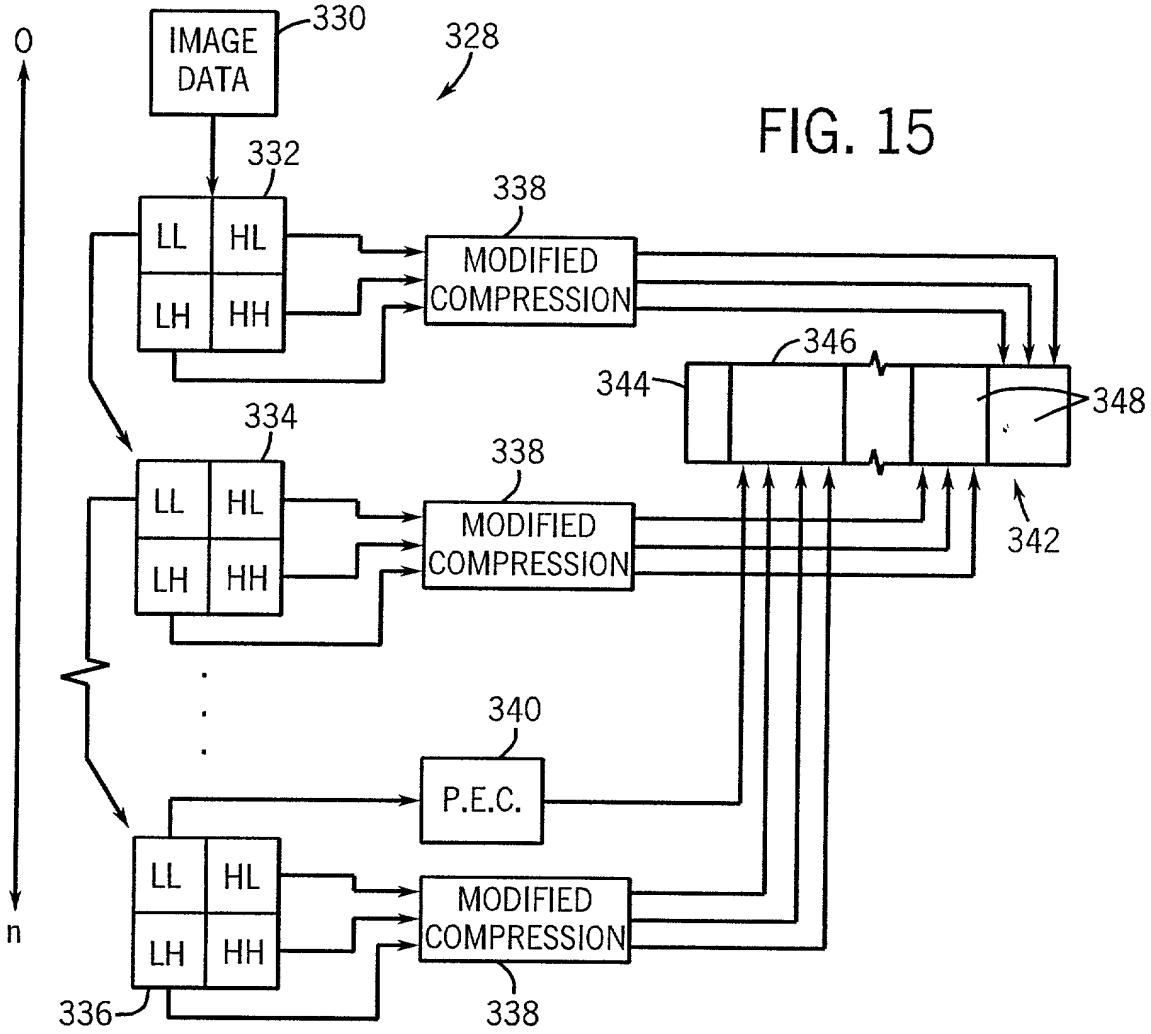

302
304
L
H
300
308
312
LL
HL
LH
HH
306
310
314
320
324
318
322
ll
hl
lh
hh
HL
326
LH
HH
316

FIG. 15


0
n
IMAGE DATA
330
328
332
LL
HL
LH
HH
338
MODIFIED COMPRESSION
346
344
348
334
338
MODIFIED COMPRESSION
342
340
P.E.C.
LL
HL
LH
HH
MODIFIED COMPRESSION
336
338

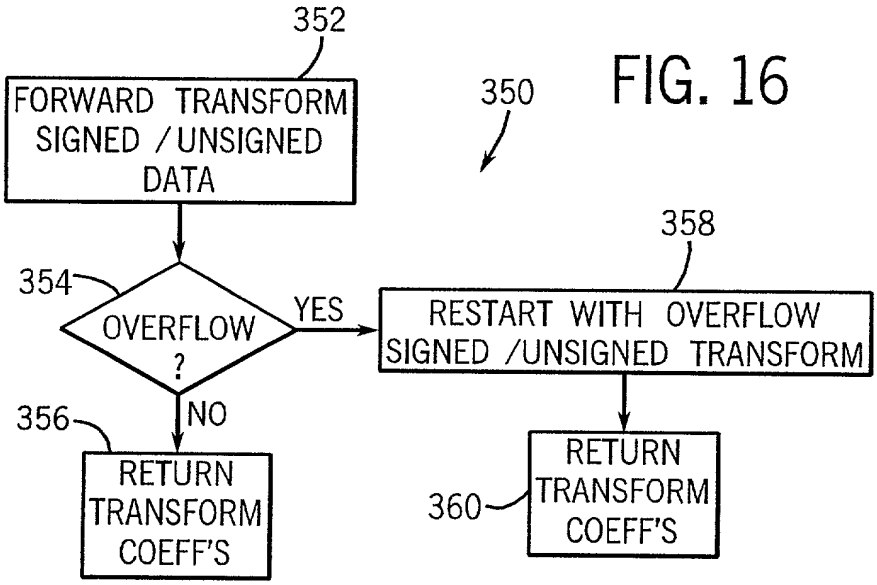

352
FORWARD TRANSFORM SIGNED / UNSIGNED DATA
350
354
OVERFLOW ?
YES
358
RESTART WITH OVERFLOW SIGNED /UNSIGNED TRANSFORM
NO
356
RETURN TRANSFORM COEFF'S
360
RETURN TRANSFORM COEFF'S

FIG. 16

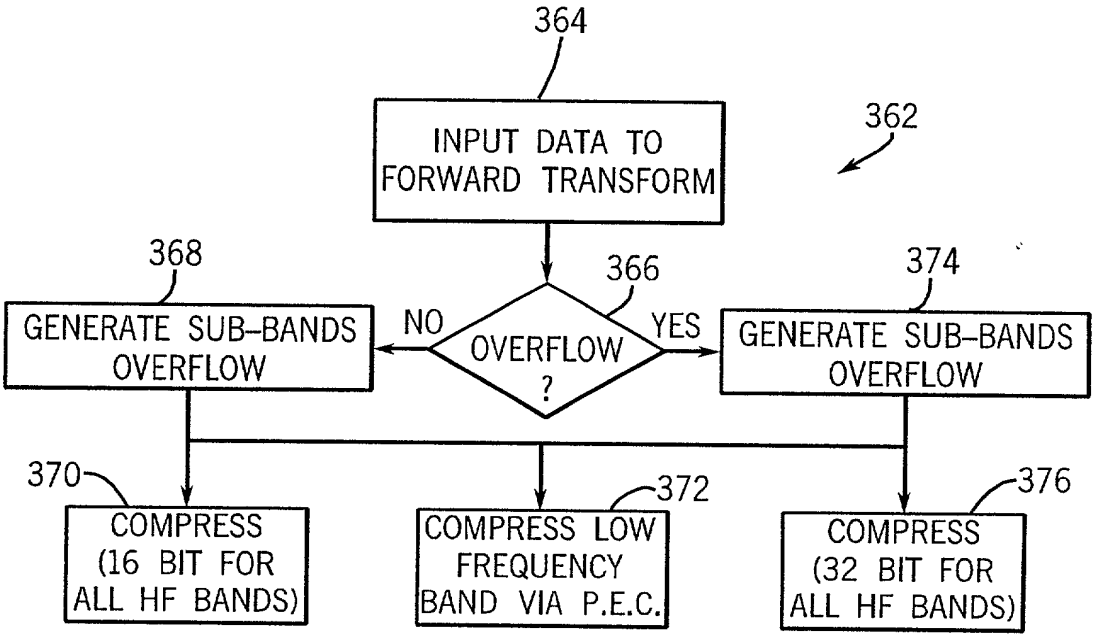
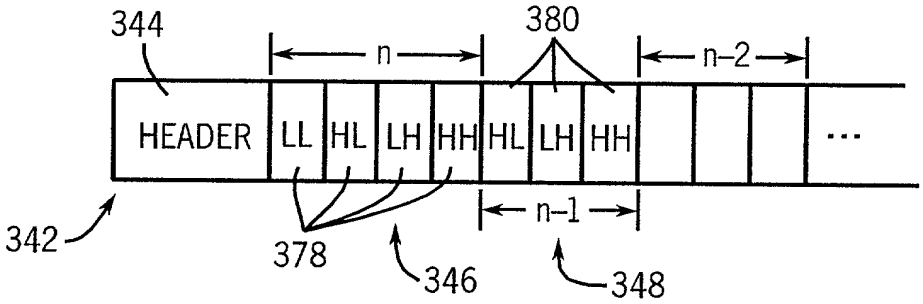

364
INPUT DATA TO FORWARD TRANSFORM
362
368
GENERATE SUB-BANDS OVERFLOW
NO
366
OVERFLOW ?
YES
374
GENERATE SUB-BANDS OVERFLOW
370
COMPRESS (16 BIT FOR ALL HF BANDS)
372
COMPRESS LOW FREQUENCY BAND VIA P.E.C.
376
COMPRESS (32 BIT FOR ALL HF BANDS)

FIG. 17

FIG. 18


344
n
380
n–2
HEADER
LL
HL
LH
HH
HL
LH
HH
...
n–1
342
378
346
348

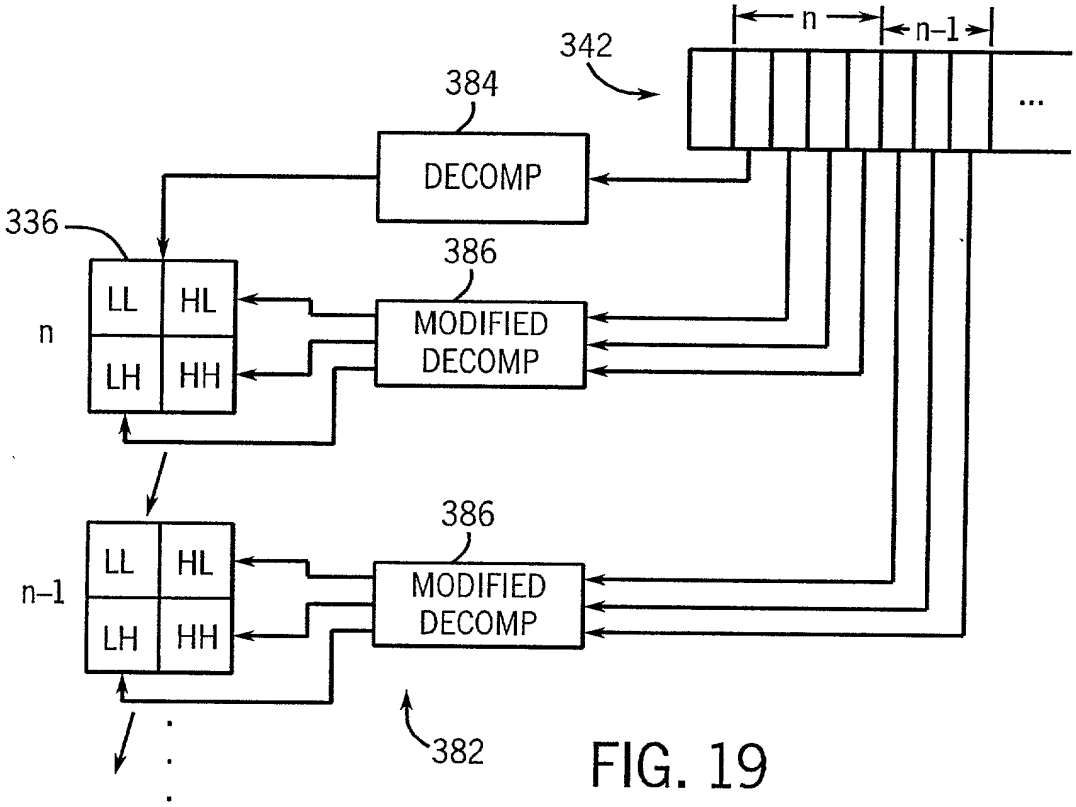

n
n–1
342
384
DECOMP
336
386
n
LL
HL
LH
HH
MODIFIED DECOMP
386
n–1
LL
HL
LH
HH
MODIFIED DECOMP
382

FIG. 19

FIG. 20

LEVEL	ROWS	COLS	COMPRESSED BYTES	EFF COMP RATIO
5	64	79	5652	1811:1
4	128	157	13780	526:1
3	256	313	49868	147:1
2	512	625	186128	40:1
1	1024	1250	724560	10:1
0	2048	2500	2929860	2.6:1

388

PATENT
DOCKET NO.: GEMS:0131

COMBINED DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As the below-named inventors, we hereby declare that:

Our residences, post office addresses and citizenship are as stated below next to our names.

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled: **IMAGE DATA COMPRESSION METHOD AND APPARATUS**, the specification of which is attached hereto.

We hereby state that we have reviewed and understand the contents of the above-identified specification, including the claim, as amended by any amendment referred to above.

We acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

We hereby claim foreign priority benefits under Title 35, U.S.C. §119 of any foreign application for patent or inventor's certificate listed below, and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

COUNTRY	APPLICATION NUMBER	DATE OF FILING (day, month, year)	PRIORITY CLAIMED UNDER 37 U.S.C. 119
None			
			☐ Yes ☐ No

We hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION SER. NO.	APPLICATION SER. NO.	FILING DATE	STATUS (patented, pending, abandoned)
None			

We hereby appoint Christian G. Cabou (Reg. No. 35,467) and Phyllis Y. Price (Reg. No. 34,234) both of GE Medical Systems, 3000 North Grandview Blvd., Waukesha, Wisconsin 53188; Ronald E. Myrick (Reg. No. 26,315), Henry J. Policinski, (Reg. 26,621), Jay L. Chaskin, (Reg. No. 24,030), James W. Mitchell, (Reg. No. 25,602), and Bernard Snyder, (Reg. No. 24,843) all of General Electric Company (W3E), 3135 Easton Turnpike, Fairfield, Connecticut 06431-0001; and Patrick S. Yoder (Reg. No. 37,479), Michael G. Fletcher (Reg. No. 32,777), Robert A. Van Someren, (Reg. No. 36,038), and Diana M. Sangalli (Reg. No. 40,798) of Fletcher, Yoder & Van Someren, 7915 FM 1960 West, Suite 330, Houston, Texas 77070, jointly, and each of them severally, our attorneys, with full power of substitution, delegation and revocation, to prosecute this application, to make alterations and amendments therein, to receive the patent and to transact all business in the Patent and Trademark Office connected therewith.

We hereby direct that all correspondence and telephone calls in connection with this application be addressed to Patrick S. Yoder, 7915 FM 1960 West, Suite 330, Houston, Texas 77070, (281) 970-4545.

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that all such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of first joint inventor: **Robert D. Barnes**

Inventor's signature: Robert D. Barnes Date: 20 Nov 00

Residence: **1383 King George Ct., Palatine IL 60067**

Citizenship: **United States of America**

Post Office Address: **1383 King George Ct., Palatine IL 60067**

Full name of second joint inventor: **Saad A. Sirohey**

Inventor's signature: S Sirohey Date: Nov. 20 '00

Residence: **1249 Deerfield Parkway #104, Buffalo Grove, Illinois 60089**

Citizenship: **Pakistan**

Post Office Address: **1249 Deerfield Parkway #104, Buffalo Grove, Illinois 60089**

Full name of second joint inventor: **Robert C. Gemperline**

Inventor's signature: Robert C. Gemperline Date: Nov. 20, 2000

Residence: **710 Chestnut Court, Algonquin, Illinois 60102**

Citizenship: **United States of America**

Post Office Address: **710 Chestnut Court, Algonquin, Illinois 60102**